VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Monitorování a správa sítí

Monitoring and Administration of Networks

2012

Bc. Tomáš Bindač

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Tomáš Bindač**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T059 Mobilní technologie

Téma: Monitorování a správa sítí.
Monitoring and Administration of Networks

Zásady pro vypracování:

Pro spolehlivý provoz sítě je velmi důležitá otázka monitorování sítě. Cílem diplomové práce je implementace systému pro monitorování a správu sítí s generováním výstupů na webových stránkách.
1. Úvod do problematiky monitorování a správy sítí.
2. Návrh řešení pro monitorování a správu síťových zařízení.
3. Implementace skriptů pro automatické generování výstupů.
4. Ověření návrhu v laboratorních podmínkách.

Seznam doporučené odborné literatury:

Barth, W.*Nagios: System and Network Monitoring*, No Starch Press 2006, ISBN-10: 1593270704
Douglas, R., Schmidt, K.*Essential SNMP, Second Edition*, O'Reilly 2005, ISBN-10: 0596008406
Dále podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Nevlud**

Datum zadání: 18.11.2011
Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA elektrotechniky a informatiky OSTRAVA

prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.*

V Ostravě 3. května 2012 .....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2012 .....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Diplomová práce se zabývá návrhem a realizací monitorovacího programu pro středně velké organizace. Ze všech open-source nástrojů byl vybrán systém Nagios, pro jeho univerzální použití. V teoretické části se zabývám obecně správou sítě, protokolem SNMP, moduly pro tvorbu grafů a popisem nejznámějších open-source programů. V praktické části se zaměřuji na konfiguraci sítě v laboratorních podmínkách, instalaci vybraného systému, jeho konfigurací a vlastními skripty.

Klíčová slova: monitoring, správa, SNMP, Nagios, skript

Abstract

Diploma thesis describes planing and implementation of a monitoring program for mid-size organizations. From all of the open-source tools Nagios system was chosen, bacause of his universal usage. In the theoretical part I'm generally occupy of a network administration, SNMP protocol, modules for charting and a description of the most popular open-source programs. In the practical part I'm focused on the network configuration in a laboratory environment, installation of selected system, his configuration and own scripts.

Keywords: monitoring, administration, SNMP, Nagios, script

Seznam použitých zkratek a symbolů

ASN.1	–	Abstract Syntax Notaiton One
ATM	–	Asynchronous Transfer Mode
BER	–	Basic Encoding Rule
CPU	–	Central Processing Unit
EGP	–	Exterior Gateway Protocol
FDDI	–	Fiber distributed data interface
GPU	–	Graphics Processing Unit
HDD	–	Hard Disk Drive
HEMS/HEMP	–	High-level Entity System/Protocol
HTML	–	Hyper Text Markup Language
HTTP	–	Hypertext Transfer Protocol
ICMP	–	Internet Control Message Protocol
IEEE	–	Institute of Electrical and Electronics Engineers
IETF	–	Internet Engineering Task Force
IGRP	–	Interior Gateway Routing Protocol
IP	–	Internet Protocol
LAN	–	Local Area Network
MAC	–	Media Access Control
MIB	–	Management Inforamtion Base
MRTG	–	Multi Router Traffic Grapher
MySQL	–	My Structured Query Language
NDOUtils	–	Nagios addon
NMS	–	Network Management System
OI	–	Object Identifier
OID	–	Object Identificator Data
PC1	–	Personal Computer 1
PC2	–	Personal Computer 2
PC3	–	Personal Computer 3
PDU	–	Protocol Data Unit
PHP	–	Hypertext Preprocessor, dříve Personal Home Page
PNG	–	Portable Network Graphics
RA	–	Router A
RB	–	Router B

RC	–	Router C
RAM	–	Random Access Memory
RFC	–	Request for Comments
RIP	–	Routing Information Protocol
RRDTool	–	Round-Robin Database Tool
SGMP	–	Simply Gateway Monitoring Protocol
SMS	–	Short Message Service
SNMP	–	Simple Network Management Protocol
SNMPD	–	Simple Network Management Protocol Deamon
SWA	–	Switch A
SWB	–	Switch B
SWC	–	Switch C
TCP	–	Transmission Control Protocol
UDP	–	User Datagram Protocol
UNIX	–	Ochranná známa operačního systému

Obsah

1 Úvod **4**

2 Teoretická část **5**
2.1 Správa sítě 5
2.2 Aktivní a pasivní monitorování 6
2.3 SNMP – Simple Network Management Protocol 7
2.4 MRTG 14
2.5 RRDTool 15
2.6 Cron 16
2.7 Používané open-source nástroje 16

3 Praktická část **21**
3.1 Konfigurace sítě 21
3.2 Instalace monitorovacího programu 21
3.3 Konfigurace monitorovacího programu 30
3.4 Monitoring (Odchytávání informací) 34
3.5 Skripty, pluginy a moduly 36
3.6 Grafy 39

4 Závěr **42**

5 Reference **44**

Přílohy **45**

A Konfigurační soubory **46**

B Skript pro kontrolu provozu na síti – check_traffic **51**

Seznam obrázků

1 Komunikace mezi agentem a managerem 9
2 Strom MIB 12
3 Webové rozhraní Cacti se správou zařízení 17
4 Webové rozhraní OpenNMS 18
5 Webové rozhraní Zabbix 19
6 Webové rozhraní Nagios 20
7 Konfigurovaná síť 22
8 Nagios NDOUtils 23
9 DataBase Configuration 27
10 User Interface Configuration 28
11 Nastavení uživatele 30
12 Nastavení skupiny uživatelů 31
13 Nastavení Templates 32
14 Nastavení Command 33
15 Nastavení Service 34
16 Export nastavení 35
17 Graf statistiky CPU 39
18 Graf statistiky CPU load 39
19 Graf statistiky ping 40
20 Graf služby check_host_alive pro zařízení SWB 40
21 Graf služby check_traffic pro zařízení RA na rozhraní s0/1/1 41
22 Graf služby ping pro zařízení RB 41
23 MRTG graf provozu na zařízení SWA 41

1 Úvod

Počítače a veškeré informační systémy a sítě s nimi spojené jsou v dnešní době nezbytnou součástí našeho života a přistupujeme k nim dnes a denně. Díky nim máme na dosah obrovské objemy informací a dat, která jsou navíc dosažitelná na téměř jakoukoliv vzdálenost. Tímto ale také rostou nároky na výkon a správu sítí jako takových. Dnešní organizace či velké firmy, a to nejen zabývající se výpočetní technikou, se neobejdou bez vlastní sítě, ať už se jedná pouze o administrace jako například účetnictví či správu dat. Bez efektivního a bezpečného chodu těchto sítí by docházelo ke ztrátám zisku těchto firem a možnému poškození jejich dobrého jména. Avšak nedílnou součástí je také bezchybnost, jelikož při špatně fungující nebo nedostupné síti je scénář naprosto stejný. Proto je velmi důležitá jejich správa a kvalitní monitoring.

Ve své diplomové práci se proto na tuto problematiku zaměřuji. Menší až středně velké organizace nemusí nutně investovat vysoké částky do komerčních systémů pro monitorování jejich sítě. Existuje mnoho open-source systémů, zabývajícími se kvalitní správou a monitoringem, které je možné pro tyto potřeby využít.

V první kapitole se věnuji teoretické části sítí, zaměřenou především na její správu. Vysvětluji zde jakými prostředky, ať už použitými protokoly či celými systémy toho dosáhnout. Dále jsem vybral několik těchto open-source systémů a stručně je popsal. Ve druhé kapitole se již zabývám vybraným nástrojem Nagios, jeho instalací na zařízení a konfigurací samotného systému. V závěru práce se pak zajímám o samotné monitorování a tvorbu grafů.

Diplomová práce je vytvořena v prostředí LaTeX, pomocí třídy *diploma*.

2 Teoretická část

Tuto práci jsem zaměřil na monitorovací systém Nagios, jelikož se jedná o velmi kvalitní a použitelný nástroj. Bohužel i přes velkou oblibu a komunitu kolem tohoto systému, není jeho instalace a konfigurace sepsána tak, aby byl ihned použitelný pro nasazení na síti. Proto jsem se ji rozhodl sepsat a věřím, že má diplomová práce pomůže jakémukoliv uživateli, ať už začínajícímu správci či laikovi, tento systém nejen nainstalovat a nakonfigurovat, ale i efektivně využívat.

2.1 Správa sítě

Správa sítě umožňuje správcům základní úlohy na jednotlivých zařízeních. V dnešní době se komunikační sítě rozrůstají co do počtu, rozsahu i složitosti. K tomu, abychom byli schopni využít veškerý jejich potenciál, je nutné použít správné řízení sítě, které musí zajistit jak funkčnost poskytovaných služeb i jejich kvalitu, tak i případné přizpůsobení se sítě požadavkům uživatelů.

Uživatelé potřebují k jejich práci fungující síť, aby ji byli schopni kvalitně a bez nepříjemných omezení, či jiných potíží vykonat. Tím pádem od správy sítě požadují pokud možno bezchybný chod, či v případě poruchy co nejrychlejší opravu. Pokud se vyskytne dlouhodobější problém, měli by být uživatelé taktéž informováni o stavu sítě s případnou dobou opravy a navrácení do funkčního stavu. Nedílnou součástí správně fungující sítě je také výkonnost, v tomto případě propustnost a doba odezvy, to vyžaduje správné monitorování veškerého stavu na síti. Nutností je také zajistit bezpečnost sítě i uživatele před neoprávněnými útoky, či vstupem. Navíc uživatelé o veškerém monitorování sítě a s tím spojený bezchybný chod sítě (například aktualizace softwaru, konfigurace, nebo třeba také defragmentace zaplněného disku, či jiné diagnostické testy), nepotřebují vědět, proto musí být zajištěna jakási průhlednost sítě. Správce naopak vyžaduje absolutní čitelnost všech zařízení v síti.

Tyto monitorovací a řídicí aplikace nejsou zapotřebí výhradně v běžném režimu, nýbrž především ve chvíli, kdy síťové prostředí neplní korektně svou funkci, nebo plní-li ji pouze částečně. Aby bylo možné zajistit správný chod řízení sítě, musí fungovat veškeré spodní vrstvy. To ovšem při vzniku kolizní situace není možné pokaždé zaručit. Pokud kupříkladu transportní protokol nebo operační systém nefungují korektně, tak nemusí být zajištěno správné kontaktování poškozeného zařízení.

V případě homogenních sítí je tato situace značně ulehčena, neboť jednotliví prodejci si sami zhotovili aplikace nebo jakési nástroje, kterými mohou být veškerá zařízení v síti sledována, monitorována jejich výkonnost a popřípadě jimi mohou být odstraněny vzniklé chyby. Problém ovšem nastává, jakmile vzrůstá velikost a obtížnost sítí, nebo jsou používaná zařízení od rozdílných výrobců tzn. síť je heterogenní. Tímto se rapidně ztěžuje správa a monitorování celé sítě. Tyto sítě jsou ve světě samozřejmě mnohem rozšířenější, proto je zde důležité zachovat soudržnost jejich jednotlivých složek. Tato heterogenita

může být zakryta pomocí velké míry abstrakce. Nejlepším příkladem takovéto heterogenní sítě je bezesporu Internet, který vznikl propojením několika fyzických sítí (např. LAN). Abstrakce v praxi znamená, že na jednotlivé sítě, z hlediska Internetu podsítě, může být nahlíženo stejně, i přesto, že každá z nich může být reprezentována pomocí jiné technologie (např. Ethernet, Frame Relay, ATM, X.25 atd.) a také může využít stejná pravidla pro přenos dat. Díky tomu se může rozlehlá heterogenní síť, jež vznikla propojením velkého počtu rozdílných sítí, zdát jako jednoduchá homogenní síť.

Ideálním řešením těchto problémů je řízení a správa přesně specifikovaná na konkrétní síť. Toto ale ve většině případů není možné, ba dokonce ani potřebné, protože záměr je použití stejného řízení a správy na různé druhy sítě. Je tedy potřeba dopracovat se ke kompatibilitě těchto nástrojů. K tomu, abychom toho byli schopni, je důležité určit jak budou přenášeny řídící informace a užití stejných způsobů vyjádření těchto informací. I když je mnoho programů pro řízení sítě, vždy zůstává zodpovědnost za veškeré řízení sítě na člověku.

Řízení sítí znamená řízení veškerých služeb a zařízení v síti. Monitoruje síťové zdroje k získání informací o výkonu a provozu na síti, provádí ukládání získaných dat k následné analýze a tímto pak diagnostikuje chyby v síti.
Každý řídící systém je tvořen těmito prvky:

- uživatelské rozhraní (k řízení dialogu s uživatelem)
- aplikace
- komunikační rozhraní
- systémové rozhraní

Avšak problémem není pouze neslučitelnost řídících systémů, ale také shodnost například uživatelského rozhraní. Jelikož jsou i řídící systémy odlišné v závislosti na dané síti, tak i správci vyžadují různá uživatelská rozhraní v závislosti na jejich specifických činnostech. Vzrůstá také potřeba přizpůsobení se jednotlivých systémů pro správu a řízení, aby byly schopny reagovat na růst sítě a stále ji správně řídit. [1]

2.2 Aktivní a pasivní monitorování

Většinu monitorovacích metod lze rozdělit mezi aktivní a pasivní. Při aktivním monitorování posíláme do sítě testovací pakety, které opět přijímáme v jiném místě sítě. Tímto způsobem můžeme měřit například zpoždění při průchodu sítí, ztrátovost nebo dosažitelnou propustnost. Nevýhodou aktivního monitorování je přidaná zátěž do sítě, zejména při měření propustnosti intenzivním datovým tokem, možné ovlivnění provozu uživatelů a to, že měříme charakteristiky našich testovacích paketů, nikoliv charakteristiky provozu uživatelů, které mohou být velmi odlišné. Je například obtížné měřit aktivně ztrátovost paketu v síti, protože ta velmi závisí na objemu a dynamice provozu, které jsou u skutečného provozu uživatelů velmi odlišné od testovacích paketů, které si můžeme

dovolit do sítě posílat.

Při pasivním monitorování neposíláme do sítě testovací pakety, ale vyhodnocujeme časové a objemové charakteristiky uživatelského provozu. Pasivní monitorování neovlivňuje uživatelský provoz a může sledovat charakteristiky, které jsou aktivním monitorováním nezjistitelné. Například jaký je objem a dynamika volné kapacity v síti, které aplikace uživatelů mají největší nároky na kapacitu sítě nebo zda v síti dochází k bezpečnostním útokům. Aktivní monitorování si lze tedy představit jako testovací sondu poslanou jednorázově nebo opakovaně do sítě, zatímco pasivní monitorování je zpravidla trvale bežící pozorovatel dění na síti.

Kromě čistě aktivního nebo pasivního monitorování jsou i metody využívající kombinace obou přístupů (vhodné například pro měření ztrátovosti), metody zpracovávající data získaná z komponentu sítové infrastruktury, např. pomocí SNMP nebo protokolu Netflow, a měření sledující stav koncové stanice. [18]

2.3 SNMP – Simple Network Management Protocol

SNMP byl zpočátku zhotoven k podpoře komunikace v propojených univerzitách a výzkumných TCP/IP sítích. Vznikl na popud chybějícího standardního protokolu, jenž je schopen podpořit velké množství různých síťových zařízení. Byl vytvořen v roce 1988 a je sadou RFC. Jednotlivé RFC definují, jak má být implementován protokol, aby vytvořil standard pro sledování a správu v Internetu.
Stejně jako TCP/IP, které vyřešilo problémy v komunikaci mezi heterogenními systémy, tak i SNMP vyřešilo zásadní problémy spojené s řízením TCP/IP sítí. SNMP samozřejmě není první řídící protokol, avšak navazuje na předchozí více či méně úspěšné předchůdce.

1. The High-level Entity Management Systém/Protocol (HEMS/HEMP) byl v roce 1987 první zkouškou řídícího protokolu. Pracovali na něm fandové Internetu. U vzniku se ale samozřejmě nebral v úvahu takovýto rozvoj Internetu, jaký známe dnes.

2. The Simply Gateway Monitoring Protocol (SGMP) se zrodil v pozdější době, jako jednoduše implementovatelný, z důvodů již zmiňovaného velkého rozvoje Internetu, avšak už nebyl tak rozvinutý a elegantní jako HEMS. Umožňuje příkazům, aby mohly být použity aplikačním protokolem pro nastavení nebo načtení hodnoty pro potřeby monitorování bran, na kterých je aplikační protokol funkční.
   Příklady monitorování:

   - Typ sítě pro rozhraní: IEEE 802.3 MAC, Ethernet, FDDI, X.25
   - Stav rozhraní: down, up, attempting, atd.
   - Typ směrování: local, remote, sub-network, atd.
   - Směrovací protokol: RIP, EGP, IGRP

Tento protokol se také neuchytil, proto bylo nutné vytvořit nový protokol na základě výsledků z HEMP a SGMP.

3. The Simple Network Management Protocol (SNMP) vznikl v letech 1988 se sadou RFC a byl vytvořen stejným týmem lidí, jako SGMP. RFC definují principy a implementaci pro protokol, jenž by vytvořil jednotný celek pro monitorování a správu v síti Internet. Architektura SGMP byla dodržena jako v případě SGMP, ale byla pozměněna syntaxe dat. Poté byly schváleny i další dokumenty k určení řídící informace SMI a k určení struktury báze řídící informaci MIB. Během několika měsíců po uvedení SNMP se potvrdilo, že SNMP nebyl pouze krátkodobým záložním řešením. U správců sítě získal velkou popularitu, proto je protokol SNMP brán jako standard pro správu sítě. Dnes se agenti SNMP stali nedílnou součástí většiny operačních systémů (Windows 98/ME, Windows NT/2000/XP, Windows Vista/7, serverů NetWare a Unix/Linux) a jsou také implementováni do většiny síťových produktů počítačového hardwaru, včetně síťových karet, routerů, switchů, hubů, bridge i tiskáren.

V následující RFC byla doplněna standardní databáze nazývaná Management Information Base (MIB), zde se definují síťové objekty. MIB má deset skupin těchto objektů (systém, rozhraní, překlad adres, IP, ICMP, TCP, UDP, EGP, přenos dat a samotný protokol SNMP). Se stále přibývajícími novými a novými funkcemi a schopnostmi je nebylo možné pomocí těchto desíti skupin objektů MIB zajistit. Proto si dodavatelé hardwaru a softwaru vyvinuli vlastní MIB. [1]

2.3.1 Verze SNMP

Následující seznam obsahuje všechny aktuální verze SNMP a stav jednotlivých IETF (Internet Engineering Task Force – zveřejnění jednotlivých RFC):

- SNMP verze 1 (SNMPv1) je aktuální standardní verze protokolu SNMP. Jedná se o plný standard IETF. Zabezpečení SNMPv1 je založeno na komunitách (communities) představujících hesla, ta jsou jednoduché textové řetězce, které umožňují každé aplikaci založené na SNMP, která zná tyto řetězce, získat přístup k MIB. Jsou zde obvykle tři možnosti společenství: *read-only*, *write-only* a *trap*.

- SNMP verze 2 (SNMPv2) je evoluce SNMPv1. Jedná se o experimentální IETF. I když je pouze experimentální, začali jej někteří dodavatelé již v praxi podporovat. Operace jako *Get*, *GetNext* a *Set*, používané v SNMPv2, jsou naprosto stejné jako ty používané v SNMPv1. Nicméně, SNMPv2 přidává a zlepšuje některé operace protokolů. Například operace *Trap* u SNMPv2 slouží stejné funkce jako u SNMPv1, ale používá jiný formát zprávy a je určen jako náhrada SNMPv1 *Trap*.

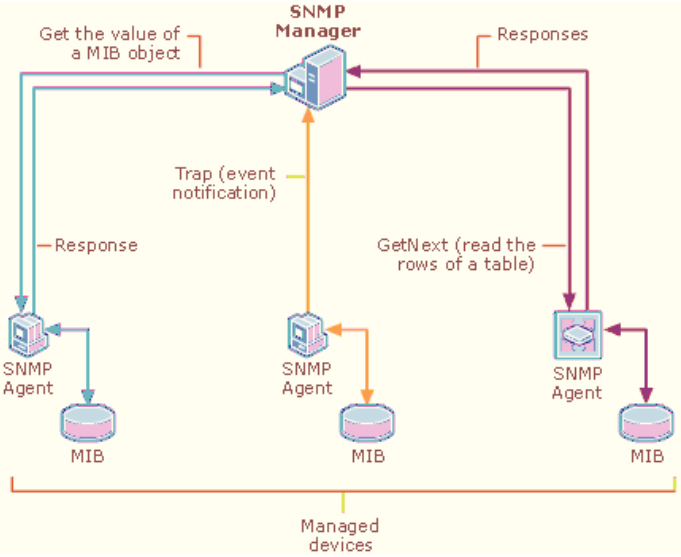
  SNMPv2 také definuje dvě nové operace: *GetBulk* a *Inform*. Operace *GetBulk* se používá k efektivnímu získávání velkého množství dat. Operace *Inform* umožňuje, aby

jeden NMS mohl poslat Trap jinému NMS, a pak přijmout odpověď. Pokud agent reagující na operaci *GetBulk* nemůže poskytovat hodnoty pro všechny proměnné v seznamu, poskytuje pouze dílčí výsledky.

- SNMP verze 3 (SNMPv3) bude další verze. V současnosti jde o navrhovaný standard. Poskytuje bezpečný přístup k zařízení pomocí kombinace ověřování a šifrování paketů v síti. Zabezpečovací model je autentizace, která je zřízena pro uživatele a skupiny, ve které jsou jednotliví uživatelé umístěni. Úroveň zabezpečení je povolená úroveň bezpečnosti v rámci bezpečnostního modelu. Kombinace zabezpečovacího modelu a úrovně zabezpečení určuje, které bezpečnostní mechanismy jsou při manipulaci s pakety SNMP použity. Jsou k dispozici tři modely zabezpečení: SNMPv1, SNMPv2c, a SNMPv3.

2.3.2 SNMP komunikace (příkazy)

Komunikace mezi jednotlivými objekty sítě se uskutečňuje na základě výměny zpráv, které jsou vidět na obrázku 1.



Obrázek 1: Komunikace mezi agentem a managerem

Na každém zařízení, neboli agentu je buď MIB databáze automaticky (routery, switche, apod.), nebo je nutné jejich MIB databázi doinstalovat, toto se týká samotných PC. Manager zasílá požadavky na agenty, kteří se dotazují do své MIB databáze a posílají zpět zprávy s odpovědí. Je také možné nastavit oznámení *Trap*, díky kterému jsou informace o vzniklé události poslány managerovi automaticky bez předchozího dotazování.

K vytvoření dotazu mezi managerem a agentem se pro správu spouští následující operace:

- *GetRequest* – Slouží k získání informací jednoho či více objektů. Tyto hodnoty jsou umístěny v tabulce MIB, kde jsou lexikograficky uspořádány. Jakmile obdrží agent dotaz *GetRequest*, určí, zda v paketu nejsou chyby, najde hodnoty v MIB a pomocí *GetResponse* odešle požadované informace managerovi zpět.
- *GetResponse* – Když jsou požadované informace vyhledány v tabulce MIB, odešlou se pomocí *GetResponse* zpět managerovi pro správu. V případě, že se v paketu objevily chyby, odešle *GetResponse* managerovi místo dat chybovou hlášku.
- *GetNextRequest* – S pomocí tohoto příkazu můžeme získat informace od jednoho či více objektů a to bez znalosti jejich přesných jmen. Je to z toho důvodu, že můžeme procházet sekvenčně tabulkami MIB, na základě pouze jediného známého objektového identifikátoru. Stanice pro správu odesílá pakety *GetNextRequest* tak dlouho, dokud nejsou přečteny všechny položky v tabulce. V případě, že nenastala žádná chyba jsou data poslána pomocí *GetResponse* opět zpět k managerovi.
- *SetRequest* – Dovoluje managerovi řídit objekty v síti pomocí úpravy jejich hodnot na agentovi SNMP. Prostřednictvím takovéto jednoduché operace je možné vytvořit nebo zrušit řízený objekt. Za předpokladu, že se nevyskytla žádná chyba je požadovaná hodnota agentem změněna a v případě úspěchu se pomocí *GetResponse* odešle její potvrzení.
- *Trap* (past) – Jedná se o jediný typ zprávy, který je vyslán agentem bez předchozího vyžádání. Stane se tak tehdy, pokud se vyskytne nějaká porucha nebo předem definovaná událost. Paket *Trap* má odlišný formát oproti všem ostatním a je pomocí UDP posílán managerovi do portu 160. Zprávy *Trap* jsou bez potvrzení, z toho důvodu nemají agenti jistotu, že byly doručeny. Jelikož můžou být tyto zprávy posílány ze spousty různých agentů, obsahují mimo jiné, hlavičku paketu OID a adresu z jakého agenta byla zpráva poslána. Existuje sedm typů těchto zpráv:
  - *ColdStart* – Znovu inicializace agenta SNMP, to umožňuje agentovi či zařízení novou konfiguraci.
  - *WarmStart* – Znovu inicializace agenta SNMP bez umožnění agentovi či zařízení nové konfigurace.
  - *LinkDown* – Agent rozpoznal chybu v připojení.
  - *LinkUp* – Spojení bylo navázáno.
  - *Authentication Failure* - Ověření agenta neproběhlo korektně.
  - *EGPNeighborLoss* – Došlo k výpadku sousedního EGP agenta SNMP.
  - *Trap EnterpriseSpecific* – Jedná se o zprávu definovanou výrobcem zařízení, která poskytuje i další informace.

Každá zpráva je zastoupena jedním datagramem, který se skládá z čísla verze, jména SNMP komunity s PDU (Protocol Data Unit - který obsahuje tělo SNMP zprávy). Zprávy SNMP nemají stanovená pevná pole, jako tomu bývá u ostatních protokolů, ale používají

podmnožinu jazyka ASN.1. Zprávy jsou přijímány na stabilních portech UDP. K popisu objektů se používá specifikace MIB a ASN.1.[1]

2.3.3 ASN.1 (Abstract Syntax Notation One)

Zde spadají data do dvou typů a to primitivní a komplexní. První jmenovaný obsahuje datové typy *Integer*, *Octet*, *String*, *Null*, *Boolean* a identifikátor *Object*. ASN.1 také umožňuje sloučit tyto primitivní datové typy do komplexního datového typu.

Samostatný objekt SNMP je vyjádřen jako identifikátor OID připojených adresou ".0" (např. OID.0 – 1.3.6.1.2.1.1.1.0). Tato přípona označuje singulární objekty, aby se jasněji odlišily od sloupcových. Sloupcové objekty mohou být například tabulky nebo řady objektů. Singulární objekt označuje pouze jeden objekt. V případě, že chceme použít více objektů, může pro podobný znak existovat mnoho datových položek. Proto v těchto případech využijeme struktur typu seznam nebo sekvencí nazývaných *tabulky*, kde každý řádek tabulky zobrazuje jeden výraz sady objektů v tabulce.

ASN.1 nabízí několik komplexních datových typů potřebných pro stavbu SNMP zpráv. Jeden komplexní datový typ, je *Sequence* (česky sekvence). Sekvence je jednoduše seznam datových polí. Každé pole v pořadí může mít jiný datový typ. ASN.1 také definuje datové typy SNMP PDU, které jsou komplexní datové typy specifické pro SNMP. Pole PDU obsahuje tělo zprávy SNMP. K dispozici jsou dva typy PDU dat *GetRequest* a *SetRequest*, které obsahují všechna potřebná data pro parametry *get* a *set*. Zpráva SNMP je struktura postavená výhradně z těchto polí datových typů ASN.1.[2]

2.3.4 Báze řídící informace MIB

Řídící informace je uvnitř systému a je nezávislá na architektuře a systému řízení. Aby bylo možné dosáhnout vzájemné propojitelnosti veškerých systémů, je potřeba, aby bylo řízení definováno vždy stejným způsobem. Veškeré uložené informace o řízení sítě jsou v MIB. Část informací uložených v MIB (informace o výkonnosti sítě, o konfiguraci, bezpečnostní parametry apod.). V určitých situacích je možné u jednoho zařízení použití více databází MIB, ty obsahují nejen informace o agentovi, ale i související informace shromážděné agentem. Například switch může obsahovat informace ohledně modelu, typu, firmwaru, diagnostiku příchozích dat, poškozených dat, cílové adresy apod. Jakmile odešle manager zprávu typu *Get*, vrátí mu agent požadované informace, nebo odešle zprávu s chybou. Při použití zprávy typu Set může manager provést změny konfigurace na řízeném zařízení. SNMP nedefinuje žádný objekt, všechny definice jsou již v MIB obsaženy.

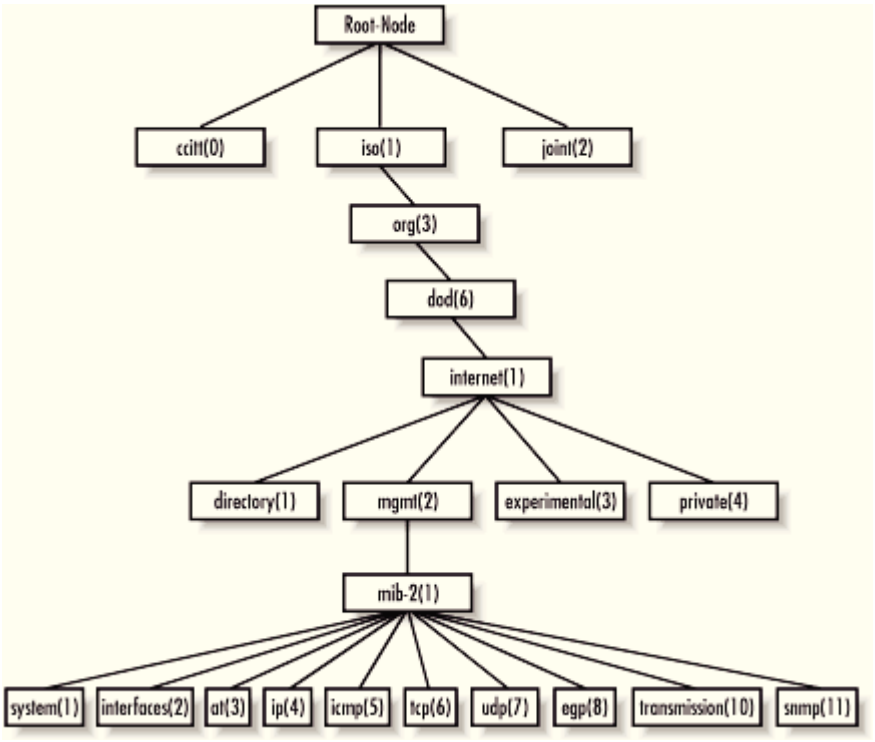
MIB je složeno ze dvou částí, a to textové (zde jsou objekty uspořádány do skupin) a modulu MIB výhradně v pojmech z ASN.1. Všechny typy objektů mají svá jména, *Object Identifier* (OI), která jsou přidělena správcem a slouží k jasné identifikaci objektu v celé MIB. OI funguje na principu logického stromu viz obrázek 2. Jednotlivé objekty v rámci SNMP jsou rozděleny do čtyř větví:

- ISO (1)
- ORG (3)
- DOD (6)
- INTERNET (1)

Avšak existuje mnoho dalších větví. Například čtyři větve po podstromu Internet:

- Podstrom *directory* (1) - rezervován pro budoucí použití OSI
- Podstrom *mgmt* (2) - zahrnuje standardy SNMP MIB I nebo II
- Podstrom *experimental* (3) - rezervován pro experimenty Internetu
- Podstrom *private* (4) - prostor pro databáze MIB jednotlivých dodavatelů

Jméno každého objektu je posloupnost jmen v hierarchii OI od kořene směrem k danému objektu (například 1.3.6.1.2.1). Množina objektů MIB II byla rozdělena na 11 kategorií.



Obrázek 2: Strom MIB

Pro veškeré objekty v síti jsou definovány úrovně přístupu. Ty určují, jestli jsou hodnoty v rámci protokolu SNMP přístupné či měnitelné pro objekty typu tabulka nebo řádek nejsou přístupné a také určují, zda jsou jednotlivé objekty implementovány povinně nebo

volitelně, popřípadě jedná-li se o zastaralý objekt. Povětšinou jsou všechny standardní objekty v MIB povinné (tzn. nesmí mít status volitelný), avšak objekty skupin TCP, UDP a EGP povinně implementují pouze agenti na příslušných prvcích sítě.

SMI i MIB Internetu nejsou závislé na určitém řídícím protokolu včetně SNMP. Ovšem nároky, které byly kladeny na SNMP (hlavně co se týče jednoduchosti) ovlivnily také definici MIB. Proto byly do MIB implementovány pouze ty nejdůležitější objekty, které byly nutné pro řízení v případě poruch a řízení konfigurace, a to pouze takové, díky kterým v případě poruchy nevznikne havárie, způsobená díky nedostatečné ochranné politice protokolu. Jako první byly vyřazeny objekty, které jsou díky vztahům s jinými objekty v MIB přebytečné. Co se týče řízených objektů do MIB, je jejich omezení u MIB I na hodnotě 100, v případě MIB II je tato hodnota zvýšena. Hlavní důraz v MIB I pro SNMP je kladen především na řízení mezisíťových počítačů (routerů) bez ohledu na zbývající prvky v síti. To se ovšem změnilo s příchodem MIB II, kde tyto informace představují zhruba 20% celkové databáze, ta má navíc přibližně dvojnásobnou velikost.

Dnes již téměř většina zařízení od výroby podporuje protokol SNMP, v případě že podporu nemají, jedná se o levnější zařízení, které je možné ovšem povětšinou nahradit použitím dražších modelů. Jde ale pouze o marketingový tah výrobců. Mezi takovéto zařízení patří například router, switch, modemy apod. Kromě toho je již vyřešeno použití SNMP pro optické sítě nebo sítě typu Ethernet. Jelikož se jedná o jednoduchý protokol, který neumožňuje takové množství řídících funkcí a operací s řídícími objekty, naproti tomu nepotřebuje komplikované agenty, proto umožňuje jednoduché zavádění. Má také nižší nároky na paměťovou a zpracovatelskou kapacitu řízených uzlů, z toho vyplývá výhoda v podobě menších nákladů za výrobky. Avšak výhoda nižších nákladů neplatí vždy, v případě dosáhnutí určité velikosti sítě náklady rapidně rostou. Proto se protokol SNMP v prvé řadě využívá v jednoduchých a málo rozsáhlých sítích, jako je například LAN (omezen na cca 1000 uzlů), kde je ale možné provádět řízení pouze v blízkosti řízených zařízení (uzlů).

2.3.5 Budoucnost SNMP

Protokol SNMP má mnoho předností. Největší je nepochybně jeho velké rozšíření a popularita. SNMP agenti jsou dostupní pro celou řadu síťových zařízení od počítačů, switchů přes routery až po modemy a tiskárny. Velká podpora ze stran výrobců i uživatelů nám ostatně sama názorně dokazuje, že je SNMP schopné vyhovět nejrozmanitějším požadavkům administrátorů na správu sítí a v jeho právu na existenci. Nadto je SNMP velice flexibilní a rozšiřitelný. Agenty je možno jakkoli rozšiřovat tak, aby bylo možné pokrýt další funkce zařízení. A právě tak lehce jde realizovat jejich upgrade po síti – nejčastěji totiž bývají nahráni ve flash pamětích síťových zařízení.

Je nutné zmínit také slabé stránky SNMP protokolu, které mohou být do budoucna šancí pro konkurenci. Některé nedostatky plynou přímo z jeho architektury, která využívá pasivních agentů, aktivních pouze na dotázání. Tato potřeba neustálého a pravidelného dotazování již může zabírat výraznou šířku přenosového pásma, které je ale primárně

určeno pro aplikace uživatelů. SNMP také nedisponuje také žádnými nástroji pro komunikaci mezi managery a funkce managementu tím pádem nemůže být šířena mezi více konzol správců. Ve větších podnicích může být použití SNMP, kvůli této omezené stupňovitosti, poněkud obtížnější. Jako další problém se také ukazuje neschopnost získat větší množství dat za pomocí jediného dotazu, například kompletní směrovací tabulka. Dalším nedostatkem je složitější zakódování dat v PDU. Protokol SNMP je také neefektivní z hlediska úspor v oblasti přenosového pásma. V každé hlavičce totiž stále přenáší tu samou informaci o verzi a stejně tak jsou v každé zprávě zbytečně kopírovány deskriptory dat.

Jako protokol je SNMP standard, který je velmi snadný a schopný značného rozšíření. Jako nástroj pro celkovou správu sítí se ale zdá poněkud komplikovaný. Pokud má správce dostatek vědomostí o komunikačních protokolech, samotné SNMP komunikaci, MIB kompilátoru, správcovské konzoli atd. má relativně výkonný prostředek pro síťovou správu. Avšak těmito znalostmi většina LAN správců nedisponuje. Je proto nutné větší standardizace SNMP pro usnadnění síťové správy, aby se neorientovaly pouze na zařízení velkých firem nebo naopak na triviální aplikace, které pracují jen s konkrétním zařízením jednoho výrobce.

I přes veškerou uvedenou kritiku je ale nutné říci, že v současnosti na trhu neexistuje pro SNMP protokol žádná konkurence určená pro efektivní správu počítačových sítí, zvláště sítí s větším rozsahem. Pro uživatelskou komunitu složitost kódování SNMP zpráv není důležitá, jelikož protokol sám o sobě je velmi efektivní a flexibilní.[5]

2.4 MRTG

Multi Router Traffic Grapher (MRTG) je nástroj pro monitorování provozního zatížení na síťových linkách. Generuje HTML stránky s PNG obrázky, které poskytují aktuální vizuální reprezentace tohoto provozu.

MRTG funguje na většině platformách UNIX a Windows NT. Je napsán v Perlu a je dodáván s plnou podporou zdroje. Využívá vysoce přenosnou SNMP implementaci napsanou v Perlu. Proto není třeba instalovat žádné další externí SNMP balíčky. MRTG také umožňuje přečíst i novou verzi (ukazatelů-counter) SNMPv2c 64bit. Tímto nevznikají žádné neshody mezi ukazateli.. Rozhraní routeru lze identifikovat pomocí IP adresy. Díky algoritmu konsolidace dat nerostou Log záznamy do obrovských velikostí. MRTG je dodáván se sadou konfiguračních nástrojů, díky nimž je konfigurace a nastavení velmi jednoduchá. Grafy jsou generovány přímo ve formátu PNG pomocí knihovny GD a vzhled webové stránky vytvořené díky MTRG je vysoce konfigurovatelný.

MRTG se skládá z Perl skriptů, které používá SNMP ke čtení provozu routerů a rychlý program napsaný v C zaznamenává údaje o provozu a vytváří grafy představující provoz na monitorované síti. Tyto grafy jsou vložené do webových stránek, které lze prohlížet z

libovolného moderního webového prohlížeče.

Kromě detailního denního zobrazení, vytváří také MRTG vizuální reprezentace provozu za posledních sedm dní, za posledních pět týdnů, a také během posledních dvanácti měsíců. To je možné díky ukládání záznamů všech dat, která ze zařízení přečetl. Tento protokol je automaticky konsolidován takže nenarůstá v průběhu času, ale ve stále konstantní velikosti obsahuje všechny podstatné údaje pro veškerý provoz z posledních dvou let. To vše je provedeno efektivním způsobem. Proto můžeme sledovat 200 nebo i více síťových zařízení.

MRTG není omezeno pouze na monitorování provozu. Je možné také sledovat jakékoliv proměnné SNMP. Můžeme dokonce použít externí program, který bude shromažďovat údaje, které měly být sledovány prostřednictvím MRTG. Správci jsou pak pomocí MRTG schopni sledovat věci, jako je zatížení systému, přihlášení, dostupnost modemu atd.. MRTG dokonce umožňuje shromáždit data ze dvou či více zdrojů do jednoho grafu.

Konfigurace
K snadnému vytvoření konfiguračního souboru lze využít skript *cfgmaker*. Například takto:

```
cfgmaker komunita@server.nekde.cz > mrtg.cfg
indexmaker mrtg.cfg > /usr/local/httpd/html/mrtg
```

Pro generování HTML stránek je vhodné zvolit adresář, ke kterému má přístup web server, tedy např. */usr/local/httpd/html/mrtg* jako v uvedeném příkladu.
Dále stačí (například pomocí cronu) v pravidelných intervalech (například každých 5 minut) spouštět příkaz

```
mrtg mrtg.cfg
```

2.5 RRDTool

Round-Robin Database Tool (RRDTool) je v informatice open-source nástroj, který se zaměřuje na zpracování a ukládání časově závislých dat (například teplota, zatížení procesoru, síťový provoz a další). Tato data jsou uložena v databázi typu round-robin, která má konstantní velikost v čase. RRDTool je nová generace nástroje MRTG, obsahuje také nástroje pro získání dat v grafické podobě. [13]

Jeho cílem je odstranit všechny možné nedostatky MRTG. Největší změnou k lepšímu je způsob ukládání a zobrazování dat. Při uložení hodnot se již automaticky zbytečně negeneruje graf. Ten si můžeme zobrazit kdykoliv jindy, a to podle přesně stanoveného období. Díky tomuto faktu ušetříme spoustu systémových prostředků. Dále také odpadly limitace počtu proměnných. Není tak problém v jednom grafu zobrazovat zároveň například vytížení serveru, obsazenou paměť a volné místo na disku. [8] Výhodou je také jeho využití v některých open-source monitorovacích programech, kde následně není nutné

ručně instalovat další dodatečné programy pro tvorbu grafů. Můžeme k nim přistupovat pohodlně z webového rozhraní a snadno si tak vykreslíme požadované grafy.

2.6 Cron

Cron, je Linux/Unix systémový nástroj, který spouští různé programy v předem definované době a intervalu (obdoba naplánovaných úloh ve Windows). Každý trochu vyspělejší webový projekt, stojící na výše jmenované platformě, se bez tohoto systémového démona neobejde. Démona Cron používají převážně Linux/Unix administrátoři ke spouštění programů a skriptů, které pomáhají udržovat funkční provoz operačního systému. Toho je dosaženo například mazáním pomocných souborů, které po sobě některé aplikace zachovají, nebo prováděním pravidelného zálohování databází či celých disků. Samozřejmě tento démon může velice dobře pomoci i tvůrcům internetových projektů. Jeho pomocí můžeme v intervalech spouštět například generování databázově velmi náročných stránek, nebo můžeme získávat aktuální informace z různých zdrojů na internetu, třeba aktuální kurzovní lístek. [7] Pro naše potřeby jej využijeme k dotazování se na zdrojové kódy pro MRTG či RRDTool a tím generování vlastních grafů v přesně určených intervalech. Grafický program je každých pět minut spuštěn Cronem, kdy pokaždé načte informace o monitorovaných zařízeních, vygeneruje graf a ukončí se.

```
*/5 * * * * <mrtg-bin>/mrtg <path to mrtg-cfg>/mrtg.cfg --logging
    /var/log/mrtg.log
```

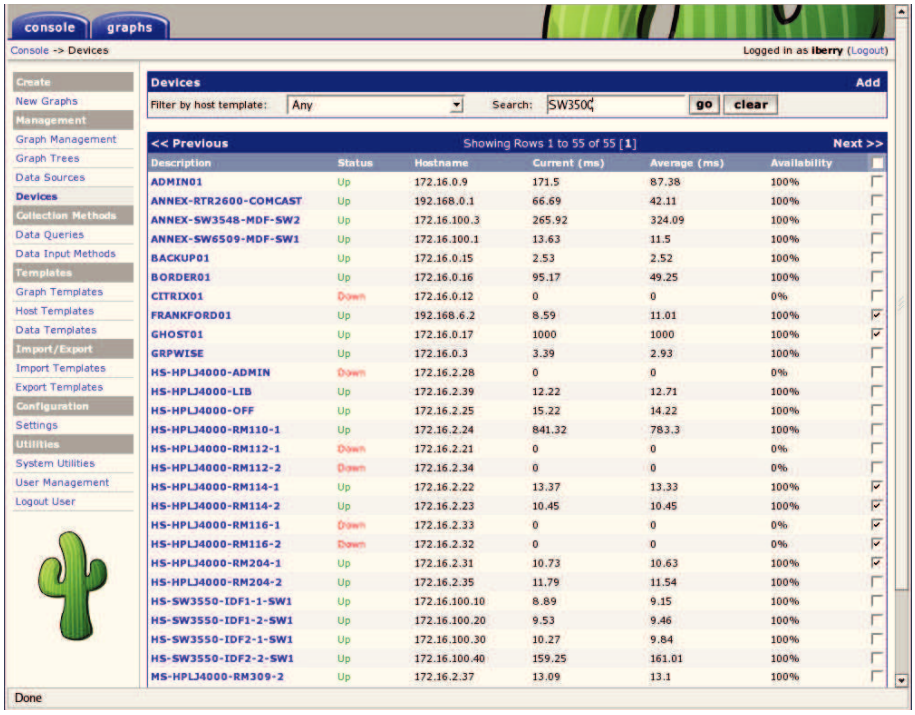
2.7 Používané open-source nástroje

V dnešní době existuje celá řadu open-source monitorovacích programů. Tyto programy jsou v hojné míře využívány v menších podnikových sítích, protože nejsou schopni, nebo nechtějí platit obrovské finanční sumy za komerční systémy. Druhou výhodou je možnost jejich použití jako zkušební/učební prostředek k monitorování sítě a případně následném zakoupení plné verze některého ze systémů. To nabízí například Nagios, který můžeme plnohodnotně využívat už ve free verzi. V případě, že ale chceme maximálně použitelný a uživatelsky přívětivější systém, můžeme zakoupit jeho plnou verzi. Pro porovnání jsem vybral pět nejpoužívanějších nástrojů, z nichž každý se liší jinými možnostmi monitoringu. Proto záleží čistě na uvážení a osobních preferencích správce sítě, který systém upřednostní.

2.7.1 Cacti

Cacti je kompletní rozhraní pro RRDTool. Round Robin Database Tool je softwarový Open Source nástroj sloužící k měření, ukládání a zobrazování strukturovaných dat ve formě grafického výstupu. Cacti ukládá veškeré potřebné informace k vytváření těchto grafů a naplňuje je daty z MySQL databáze. Je komplexním monitorovacím systémem tvořeným v PHP. Spolu se schopností spravovat v databázi grafy a zdrojová data, dokáže Cacti

data také sbírat. Pro ty, kteří používají ke tvorbě grafů MRTG, má také podporu SNMP protokolu.



Obrázek 3: Webové rozhraní Cacti se správou zařízení

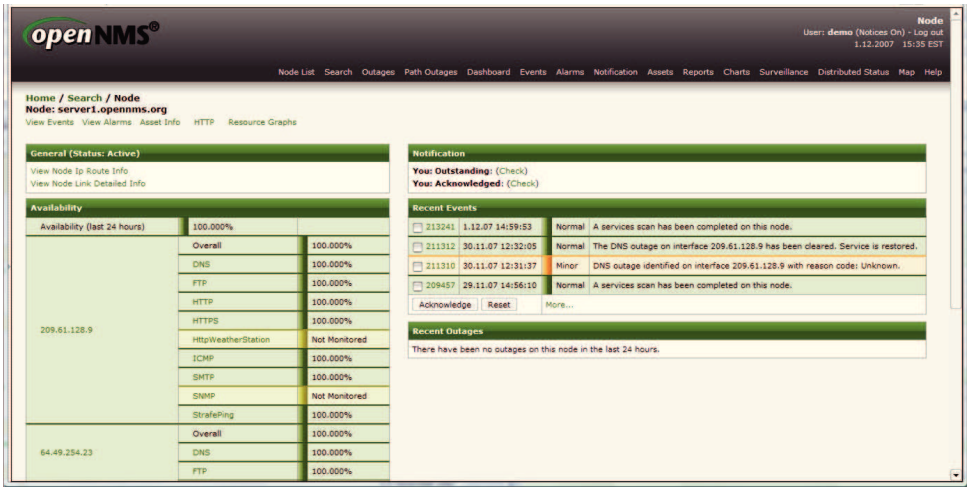
Cacti umožňuje spravovat uživatele, kterým lze takto zpřístupnit jen důležité grafy a informace. Od základu byl navržen pro monitorování stovek zařízení. K samotné práci využívá převážně SNMP protokol, je však možné vyrobit si vlastní skripty a skrze ně předávat Cacti informace. Za zmínku stojí možnost monitorovat dostupnost pingem, číst libovolné informace přes SNMP (např. vytížení zařízení, datový tok, stav tiskáren, routerů, serverů, měřit teplotu a další uživatelsky definovatelné položky). Systém můžeme nasadit do rozsáhlého prostředí, protože zvládá sledovat stovky až tisíce hodnot. [3]

Hlavní výhodou Cacti je plně grafický režim, tudíž veškerá monitorování a administrace a správa lze provádět v přehledném grafickém rozhraní viz obrázek 3. Existuje silná komunita uživatelů tohoto systému z řad administrátorů, kteří neustále vyvíjejí nové pluginy, proto jej můžeme doplnit o řadu nadstandardních služeb.

2.7.2 OpenNMS

OpenNMS umí generovat vlastní události, popř. události z odchozích zdrojů, jako například *SNMP Traps*, *Syslog* atd. Může sloužit jako centrální uložiště pro síťové události. Je schopen zvládat tisíce událostí za sekundu. OpenNMS má také řadu korelačních metod

pro automatické čištění událostí, jejich překladu na jinou událost a také dokáže redukovat duplicitní událost do jediné. Samozřejmostí je odesílání chybových oznámení formou SMS, či emailu. OpenNMS je napsán v jazyku Java, je proto dostupný na většině distribucí Linuxu, Windowsu, Solarisu a OS X. Velká výhoda celého systému oproti jiným open-source monitorovacím nástrojům spočívá v automatickém prozkoumání sítě a nastavení služeb podle jejích potřeb. Tento software je vytvořen pro bezchybnou funkčnost v široké řadě síťových prostředí. OpenNMS má velmi aktivní komunitu uživatelů. Pomocí distribuovaného monitoringu na více serverech dokáže, například oproti Nagiosu, monitorovat i desetitisíce zařízení. Opět je zde možnost doplnit stávající software o systém pluginů a tím dosáhnout plnohodnotnějšího monitoringu.



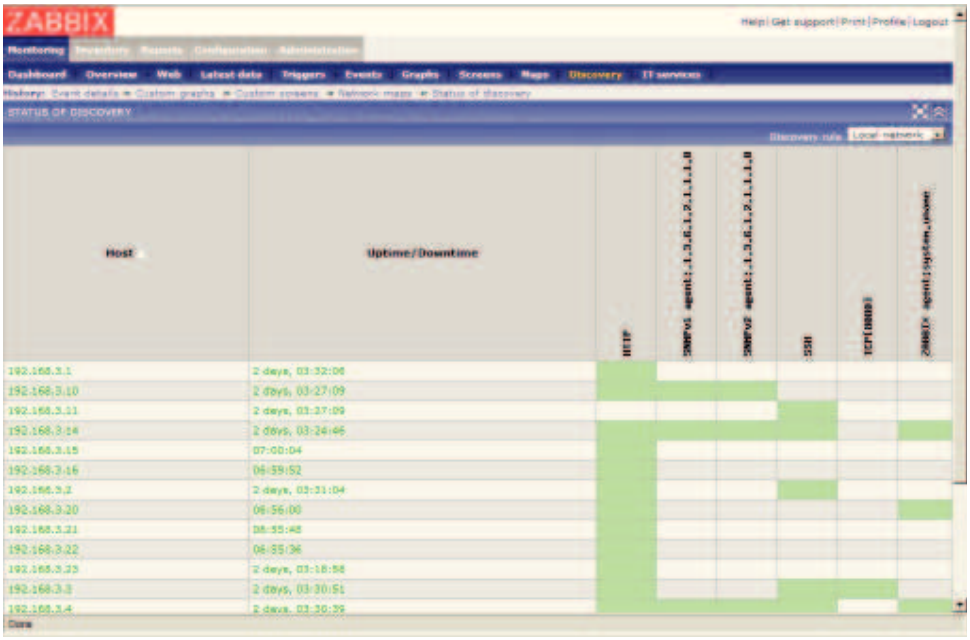
Obrázek 4: Webové rozhraní OpenNMS

Webové rozhraní na obrázku 4 vypadá vzhledově velice hezky, bohužel je příliš jednoduché a nedisponuje tak rozsáhlými možnostmi, jako v případě jiných produktů.

2.7.3 Zabbix

Zabbix je určen ke sledování stavu různých síťových služeb, serverů a dalších zařízení. Je napsán v programovacím jazyce C a jeho webové rozhraní v PHP. Zabbix nabízí jednoduché možnosti sledování sítě, jako kontrolu a ověření dostupnosti standardních služeb SMTP nebo HTTP bez nutnosti instalace softwaru na sledovaném počítači. Zabbix agent může být nainstalován na počítači jak s operačním systémem UNIX, tak i Windows a zde poskytuje informace o zatížení procesoru, využití sítě, místa na disku, apod. Je schopen monitorovat dostupnost a obsah webových stránek. Veškerá data jsou uložena v relační databázi. Abychom nemuseli na hostovi instalovat agenta, můžeme využít monitorování pomocí SNMP protokolu. Od verze 1.4 podporuje funkci autodetekce zařízení v síti, neboli autodiscovery.

Nevýhoda systému je v jeho uzavřenosti vůči dalším pluginům. Systém obsahuje pouze základní funkce pro monitoring pomocí SNMP, vizualizaci pomocí map a jednoduché upozornění na stav zařízení (online – offline). Pro menší sítě je však tento systém, díky jeho snadné instalaci a správě, plně dostačující.



Obrázek 5: Webové rozhraní Zabbix

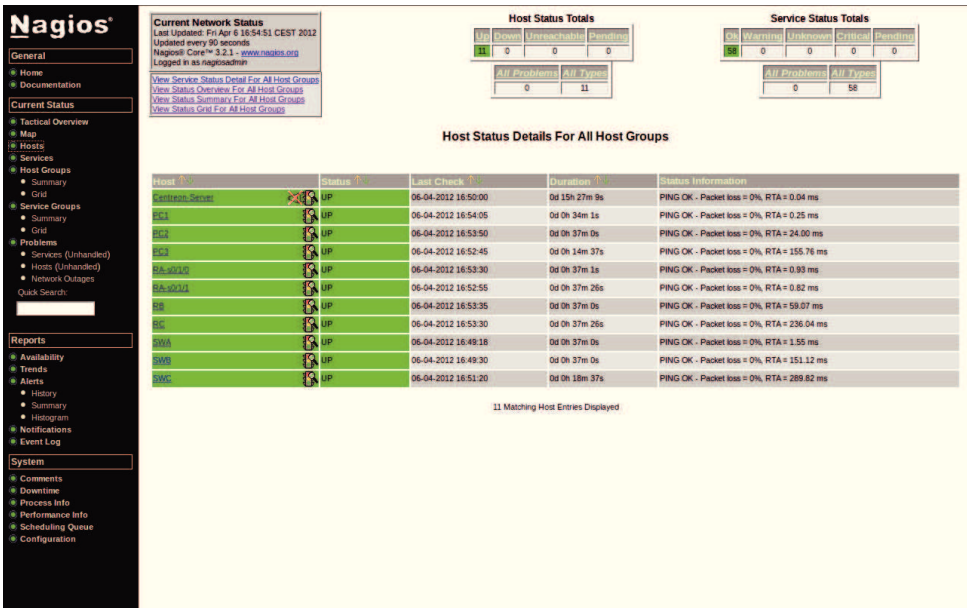
Na obrázku 5 vidíme webové rozhraní systému Zabbix, IP adresy jednotlivých hostů a jejich stav.

2.7.4 Nagios

Nagios je nástroj pro sledování systému. To znamená, že neustále kontroluje stav zařízení a jejich služby. Hlavním cílem systémového monitoringu je odhalit a podat zprávu o jakékoli chybě na síti, co možná nejrychleji tak, abychom si byli vědomi problému ještě dříve, než jej zjistí uživatel. Nagios démon, viz obrázek 6, neprovádí žádné kontroly hostů ani služeb. Ke skutečné kontrole je nutné doinstalovat pluginy (zásuvné moduly) zajišťující monitorovací a testovací funkce. To z něj činí velmi modulární a flexibilní řešení pro monitorovaná zařízení.

Objekty sledované Nagiosem jsou rozděleny do dvou kategorií: *hosts* (počítače) a *services* (služby). Hosti jsou fyzické stroje (servery, routery, pracovní stanice, tiskárny atd.), naproti tomu služby jsou určité funkce, jako například webový server, který lze definovat jako službu, která má být monitorována. Každá služba je propojena s hostem, na kterém je spuštěna. Kromě toho hosti a služby mohou být seskupeny do skupin hostů a služeb.

Nagios provádí všechny kontroly pomocí pluginů. Jedná se o externí součásti díky nimž Nagios získává informace o tom, co by mělo být kontrolováno a jaké jsou limity pro meze warning a critical. Pluginy odpovídají za kontrolu a analýzu výsledků. Výstup z těchto kontrol je status (OK, WARNING, CRITICAL, nebo UNKNOWN) a další text poskytuje podrobné informace o službě. Tento text je určen především pro správce systému, aby byli schopni zjistit podrobné statusy služeb.[6]



Obrázek 6: Webové rozhraní Nagios

Hlavní síla Nagiosu spočívá v jeho flexibilitě. Může být konfigurován přesně tak, jak daná infrastruktura sítě vyžaduje. Obsahuje mechanismy pro automatickou reakci na vzniklý problém a také velmi dobrý systém chybových oznámení. Celý tento systém je založen na čistě objektové definici několika typů objektů, které budou podrobněji vysvětleny níže od kapitoly 3.3.1.

2.7.5 Centreon

Centreon byl zpočátku vyvíjen pouze jako konfigurační rozhraní k Nagiosu. Díky postupnému přidávání základních technických prvků a služeb se z něj časem stal plnohodnotný nástroj pro monitorování sítě. Centreon dnes nabízí všechny funkce, které jsou nezbytné pro plně profesionální dohled. Je založen na jádru Nagios. Veškerá nastevní a získaná data ukládá do MySQL databáze a do konfiguračních souborů Nagiosu. Centreon podporuje všechny služby jako standardní Nagios, navíc umožňuje zobrazit 3D mapu sítě, či zobrazovat mapy pomocí programu RRDTool. Obrázky samotného systému Centreon jsou zobrazeny níže od kapitoly 3.3.1.

3 Praktická část

V praktické části se zaměřuji na vlastní hardwarové propojení a konfiguraci sítě v laboratorních podmínkách. Poté provedu krok za krokem instalaci monitorovacího softwaru a jeho konfiguraci společně s instalací dodatečných pluginů a jejich výstupem v podobě grafů služeb na síti.

3.1 Konfigurace sítě

Pro názornou ukázku funkčnosti monitorovacího systému jsem vytvořil v laboratorních podmínkách síť, viz obrázek 7.

Snažil jsem se vybrat aktivní prvky sítě s ohledem na to, aby bylo dosaženo co možná nejlepší shody s reálnými sítěmi, a také s přihlédnutím na vybavení laboratoře. Tato síť obsahuje tři routery od společnosti Cisco (Router Cisco 2800 series) pojmenované RA, RB a RC, které jsou mezi sebou propojeny sériovými kabely. Místo nich může být v reálném provozu WAN síť. Dále byly ke každému routeru připojeny Cisco switche (Cisco C2960), pojmenované SWA, SWB a SWC. Ke každému switchi byl připojen minimálně jeden počítač, na obrázku PC1, PC2 a PC3. Vlastní monitorovací server Nagios byl nainstalován na notebooku Acer Aspire 2920Z a připojen ke switchi SWA pod IP adresou 172.16.0.4.

Instalace probíhala na notebooku s operačním systémem Ubuntu 10.10 Maverick. Specifikace notebooku:

- Acer Aspire 2920Z
- CPU Dual-core 1,73GHz
- GPU Intel X3100 358MB
- RAM 2GB DDR2
- HDD 160GB

Všechny konfigurační soubory pro routery, switche a jednotlivá PC jsou ke shlédnutí v příloze A.

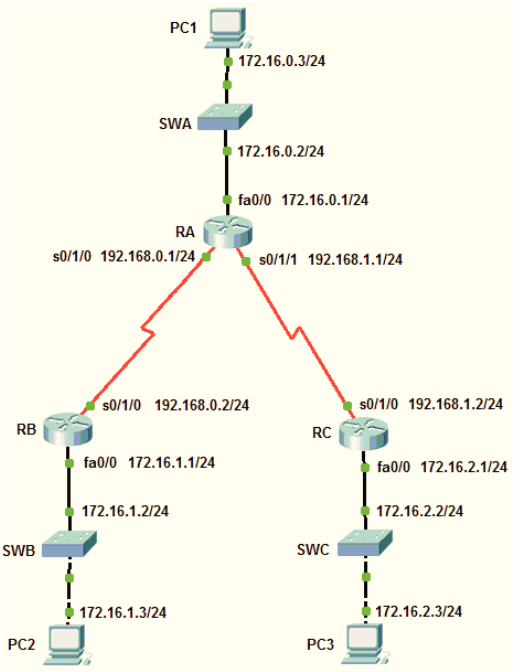
3.2 Instalace monitorovacího programu

3.2.1 Instalace a nastavení doplňkových balíků

Před samotnou instalací Nagiosu je nutné nejprve nainstalovat a nastavit jednotlivé doplňkové balíky, díky kterým bude systém vykonávat plnohodnotný monitoring dané sítě.

Nejprve si nainstalujeme emailový klient pro příjímání chybových událostí na síti

```
apt-get install postfix
```



Obrázek 7: Konfigurovaná síť

Apache server vytváří a spravuje open-source HTTP server pro moderní operační systémy zahrnující UNIX i Windows NT

```
apt-get install apache2 apache2-mpm-prefork
```

PHP5 s jeho dodatečnými balíky

```
apt-get install php5 php5-gd php5-ldap php5-mysql php5-snmp php-
    pear
```

MySQL Server[1]

```
apt-get install mysql-server-5.0 libmysqlclient15-dev
```

SNMP a SNMPD

```
apt-get install snmp snmpd libsnmp-perl libnet-snmp-perl
```

Instalace RRDTool, který slouží k získávání, zpracování a ukládání časově závislých informací, jako například vytížení procesoru apod., které následně dokáže zobrazit v grafické podobě.

```
apt-get install rrdtool librrds-perl
```

[1]Zde budeme dotázání k vytvoření nového hesla pro *root* uživatele MySQL, toto heslo bude potřebné v následující instalaci.

Perl moduly, jedná se o určitý soubor konvencí pro použití programovacího jazyku Perl a jeho balíků.

```
apt-get install libconfig-inifiles-perl libdigest-hmac-perl
   libdigest-sha1-perl libcrypt-des-perl libgd-gd2-perl
```

3.2.2 Instalace samotného Nagiosu

Zde nastavíme heslo pro uživatele *nagiosadmin*

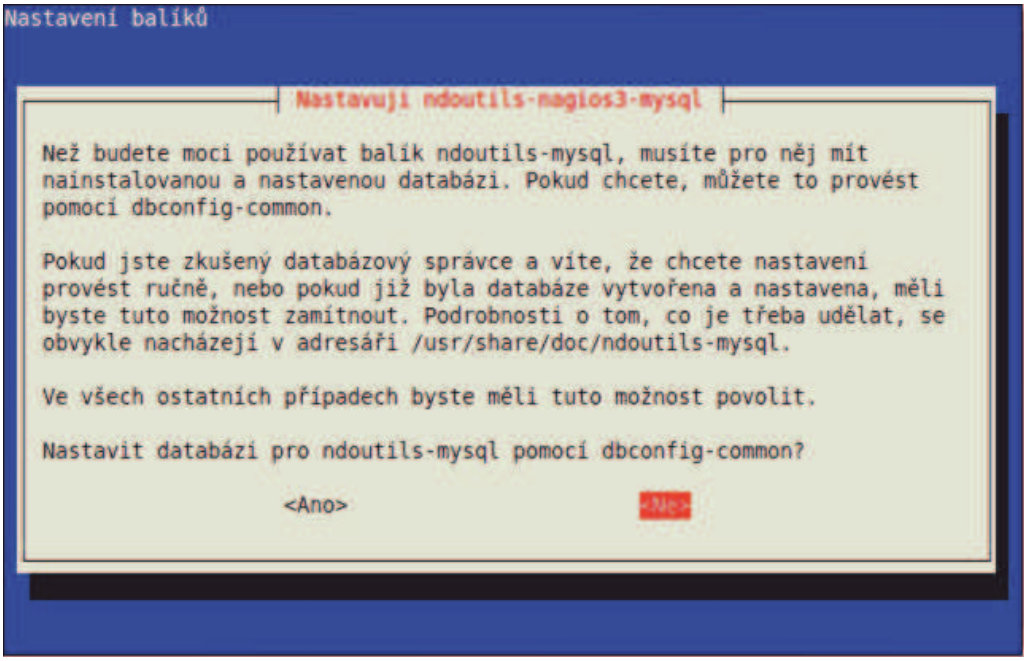
```
apt-get install nagios3
```

Doplňující balík Nagios plugin

```
apt-get install nagios-nrpe-plugin
```

Balík Nagios MySQL

```
apt-get install ndoutils-nagios3-mysql
```



Obrázek 8: Nagios NDOUtils

Zde se nás instalátor zeptá, jestli chceme nastavit databázi pro MySQL automaticky. Vše nastavíme ručně, proto vybereme možnost *NO*, viz obrázek 8

Pro jistotu si zálohujeme vytvořený konfigurační adresář Nagiosu, jelikož webové rozšíření Centreon si vytvoří tento adresář znovu.

```
mv /etc/nagios3 /etc/nagios3.orig
```

Vytvoříme opět nový adresář nagios3 v */etc*

```
mkdir /etc/nagios3
```

Zkopírujeme

```
cp -Rt /etc/nagios3 /etc/nagios3.orig/nagios.cfg /etc/nagios3.orig
   /apache2.conf /etc/nagios3.orig/stylesheets/
```

Nastavení práv pro Nagios

```
chown nagios:www-data /etc/nagios3
chmod ug+w /etc/nagios3
```

3.2.3 Instalace rozšíření Centreon

Ze stránek *http://www.centreon.com/Home-Download/download-home* si stáhneme nejnovější verzi programu, doporučoval bych vybírat stabilní verze (Stable version). V mém případě se jednalo o verzi centreon-2.3.4.

Rozbalíme balík centreon-2.3.4, v případě, že se jedná o nejnovější verzi, tak příkaz upravíte na tuto verzi

```
tar xzf centreon-2.3.4.tar.gz
```

Přejdeme do rozbalené složky

```
cd centreon-2.3.4
```

Spustíme instalační skript

```
./install.sh -i
```

Po přečtení licenčních ujednání (odklikneme mezerníkem) pokračujeme dle následujících dotazů. Ve většině případů se jedná o defaultní nastavení, avšak upozorňuji, že některé cesty k adresářům se musí změnit. Zde jsou zobrazeny pouze ty hodnoty, které bylo nutno změnit. Ostatní hodnoty budou v defaultním stavu, proto je odklikneme Entrem. V případě dotazů *yes/no* zadáváme vždy *yes (y)*.

Klikneme na Enter k přečtení Centreon License, podržením mezerníku sjedeme až na konec, potom pokračujeme v instalaci.

```
1  ------------------------------------------------------------
2    Please choose what you want to install
3  ------------------------------------------------------------
4
5    BEZE ZMĚN
6
7  ------------------------------------------------------------
8    Start CentWeb Installation
9  ------------------------------------------------------------
10 Where is your Centreon log directory
11 default to [/usr/local/centreon/log]
12 > /var/log/centreon
13
14 Do you want me to create this directory ? [/var/log/centreon]
15 [y/n], default to [n]:
16 > y
17 Path /var/log/centreon                        OK
18
19 Where is PEAR [PEAR.php]
20 default to [/usr/share/php/PEAR.php]
21 >
22 Path /usr/share/php                           OK
23
24 Where is installed Nagios ?
25 default to [/usr/local/nagios]
26 > /usr/lib/cgi-bin/nagios3
27 Path /usr/lib/cgi-bin/nagios3                 OK
28
29 Where is your nagios config directory
30 default to [/usr/local/nagios/etc]
31 > /etc/nagios3
32 Path /etc/nagios3                             OK
33
34 Where is your Nagios var directory ?
35 default to [/usr/local/nagios/var]
36 > /var/lib/nagios3
37 Path /var/lib/nagios3                         OK
38
39 Where is your Nagios plugins (libexec) directory ?
40 default to [/usr/local/nagios/libexec]
41 > /usr/lib/nagios/plugins
42 Path /usr/lib/nagios/plugins                  OK
43 /usr/sbin/nagios3                             OK
44
45 Where is your Nagios image directory ?
46 default to [/usr/local/nagios/share/images/logos]
47 > /usr/share/nagios/htdocs/images/logos
48 Path /usr/share/nagios/htdocs/images/logos    OK
```

```
/usr/sbin/nagios3stats                          OK
p1_file : /usr/lib/nagios3/p1.pl                OK
/usr/bin/php                                    OK
/usr/bin/perl                                   OK
Finding Apache group :                          www-data
Finding Apache user :                           www-data
Finding Nagios user :                           nagios
Finding Nagios group :                          nagios

------------------------------------------------------------
  Configure Sudo
------------------------------------------------------------

  BEZE ZMĚN

------------------------------------------------------------
  Configure Apache server
------------------------------------------------------------

  BEZE ZMĚN

------------------------------------------------------------
Pear Modules
------------------------------------------------------------

beze změn, pozor, nelekneme se červených výsledků NOK, vše je v
    pořádku, zde je vše v pořádku

------------------------------------------------------------
  Start CentStorage Installation
------------------------------------------------------------

  BEZE ZMĚN

------------------------------------------------------------
  Start CentCore Installation
------------------------------------------------------------

  BEZE ZMĚN

------------------------------------------------------------
  Start CentPlugins Installation
------------------------------------------------------------

  BEZE ZMĚN

------------------------------------------------------------
  Start CentPlugins Traps Installation
------------------------------------------------------------
```

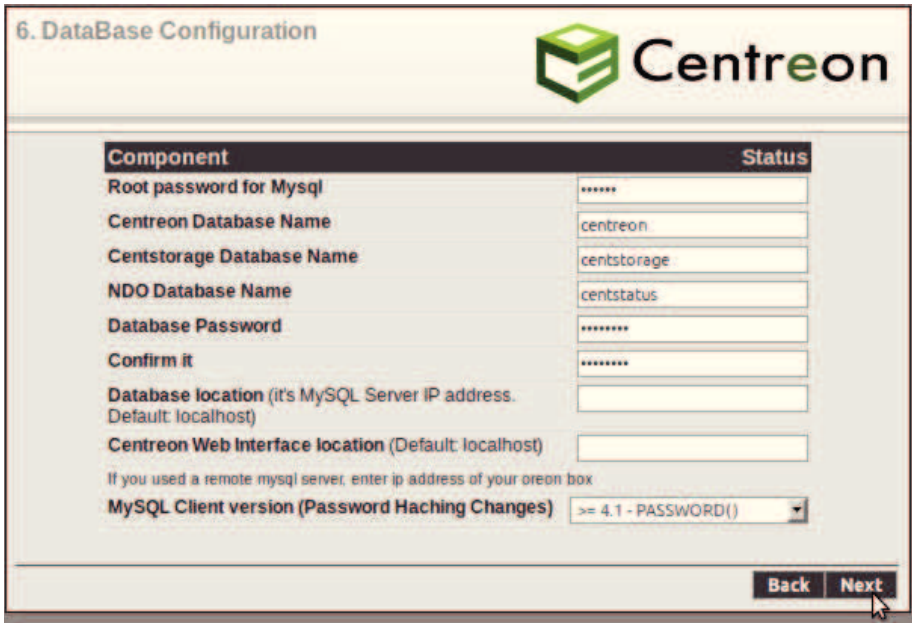
```
97
98    BEZE ZMĚN, zde je instalace u konce
```

3.2.4 Instalace Centreonu ve webovém rozhraní

Spustíme si libovolný webový prohlížeč a zadáme http://<*IP_serveru*>/*centreon* . V mém případě vypadá adresa následovně *http://127.0.0.1/centreon*

Zde na nás čeká 12 obrázků:

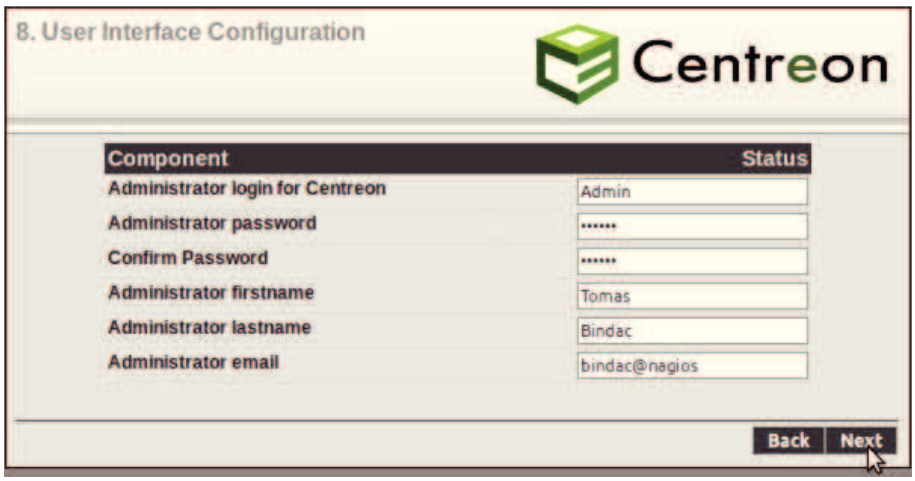
1. Welcome to Centreon Setup – *Start*
2. Licence – *I Accept* + *Next*
3. Environment Configuration – Nastavíme Nagios Version na *3.x* + *Next*
4. Verifying Configuration – *Next*
5. Verifying PHP Pear Component – *Next*
6. DataBase Configuration – Zde zadáme své heslo pro MySQL databázi, dále pokračujeme dle obrázku 9, heslo nastavíme na centreon, potvrdíme a jako poslední zaklikneme MySQL klienta na verzi *4.1* + *Next*



Obrázek 9: DataBase Configuration

7. DataBase Verification – *Next*

8. User Interface Configuration – Zde zadáme údaje o administrátorovi, například viz obrázek 10 – *Next*



Obrázek 10: User Interface Configuration

9. LDAP Authentication – Vybereme *No* + *Next*
10. Centreon Configuration File – *Next*
11. Creating Database – *Next*
12. Post-Installation – *Click here to complete your install*

3.2.5 Konfigurace NDOUtils

Od verze Centreonu 2.1 a výš je databáze NDOUtils definovaná automaticky, je ji pouze nutné potvrdit v konfiguračním souboru */usr/default/ndoutils*. Upravíme ho tedy příkazem

```
vim /usr/default/ndoutils
```

zde změníme hodnotu ENABLE_NDOUTILS=0 na ENABLE_NDOUTILS=!1

3.2.6 Finální konfigurace Centreonu

Připojíme se k webovému rozhraní Centreonu, opět pomocí *http://127.0.0.1/centreon* a zadáme uživatelské jméno *Admin* a heslo, které jsme si zvolili v předchozí části instalace webového rozhraní

Přejdeme na panel *Configuration* dále *Nagios*, v levém sloupci menu vybereme položku *cgi* a klikneme na odkaz *CGI.cfg*. Zde upravíme následující položky:

```
- Physical HTML Path: /usr/share/nagios3/htdocs
- URL HTML Path : /nagios3
- Nagios Process Check Command: /usr/lib/nagios/plugins/check-
    nagios /var/cache/nagios3/status.dat 5 '/usr/sbin/nagios3'
```

Uložíme tlačítkem *Save*.

V menu vybereme položku *nagios.cfg* a klikneme na *Nagios CFG 1*, kde v panelu *Files* upravíme následující položky:

```
- Log File : /var/log/nagios3/nagios.log
- Downtime File : /var/lib/nagios3/downtime.dat
- Comment File : /var/lib/nagios3/comment.dat}
- Temp File : /var/cache/nagios3/nagios.tmp}
- P1 File : /usr/lib/nagios3/p1.pl}
- Lock File : /var/run/nagios3/nagios3.pid}
- Object Cache File : /var/cache/nagios3/objects.cache}
- Status File : /var/cache/nagios3/status.dat}
- External Command File : /var/lib/nagios3/rw/nagios.cmd
```

V sekci *Logs Options* nastavíme

```
- Log Archive Path : /var/log/nagios3/archives/
- State Retention File : /var/lib/nagios3/retention.dat
```

Opět uložíme tlačítkem *Save*

Přejdeme na panel *Administration* dále *Options*, v levém sloupci vybereme *CentStorage* a nastavíme

```
- Nagios current log file to parse: /var/log/nagios3/nagios.log
```

Uložíme tlačítkem *Save*.

V panelu *Configuration* dále *Nagios* zkompilujeme celou konfiguraci tím, že zaškrtneme veškeré položky v jeho okně a v roletce *Method* zvolíme *External Command*. Pak už jen klikneme na *Export* a vše se nakopíruje do složky */etc/nagios*. Aby se nám nastavení z Centreonu promítlo do Nagiosu, musíme takto postupovat při jakékoliv změně nastavení, či monitorování.

3.2.7 Dokončení instalace

K nastartování démona centreon a centstorage je zapotřebí v distribuci Ubuntu/Debian zadat následující příkaz, který nastaví shell uživatele nagios (na hodnotu true)

```
usermod -s /bin/sh nagios
```

Aby bylo možné spouštět externí příkazy, musíme nastavit práva pro webové rozhraní Nagiosu

```
invoke-rc.d nagios3 stop
dpkg-statoverride --update --add nagios www-data 2710 /var/lib/
    nagios3/rw
dpkg-statoverride --update --add nagios nagios 751 /var/lib/
    nagios3
```

Přidáme uživatele *nagiosadmin* a nastavíme mu heslo například na *Pass*

```
htpasswd -bc /etc/nagios3/htpasswd.users nagiosadmin Pass
```

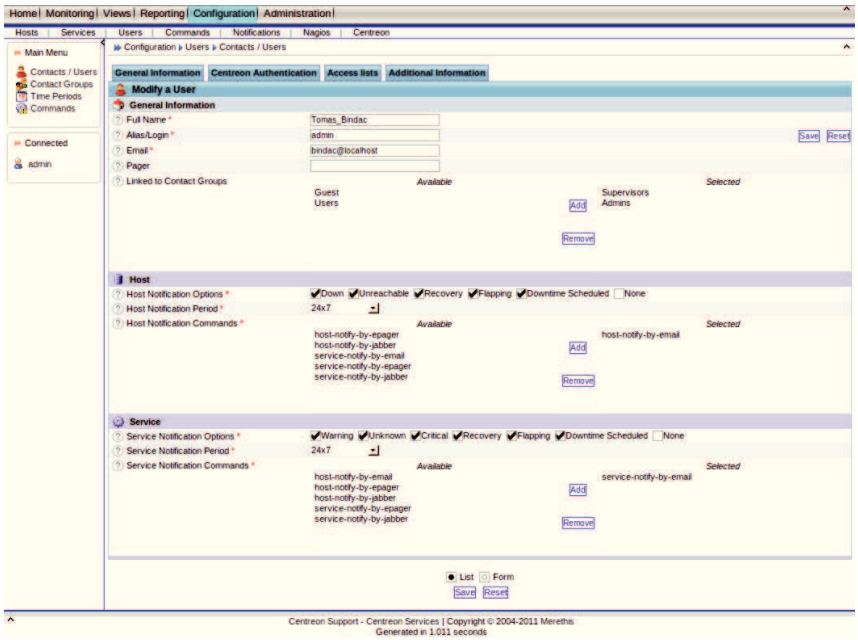
Nyní už stačí pouze restartovat server a je instalace dokončena

```
sync;sync;init 6
```

3.3 Konfigurace monitorovacího programu

V rozhraní Centreonu vybereme panel *Configuration*, objeví se nám podpanel, kde si můžeme vybrat následující položky:

3.3.1 Contact



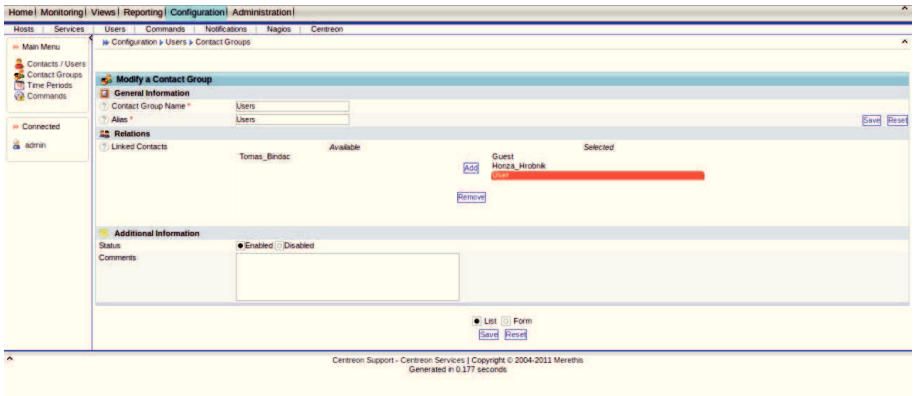
Obrázek 11: Nastavení uživatele

Nejprve je nutné definovat uživatele, kde nastavíme jeho jméno, práva, email (sms, aim, atd.) a jak často má být informován v případě výpadku. Na obrázku 11 jsem vybral

uživatele *Tomas_Bindac*, který má emailovou adresu *bindac@localhost* a patří do skupiny Supervisors a Admins. Pro hosty je nastaveno odesílání všech typů chybových zpráv (*down* - mimo provoz, *unreachable* - nedostupný, *recovery* - oživení, *flapping* - nestálý a *downtime scheduled* - plánované odstavení), které budou odesílány 24 hodin 7 dní v týdnu, formou emailových zpráv. Pro služby je nastavení stejné.

3.3.2 Contactgroups

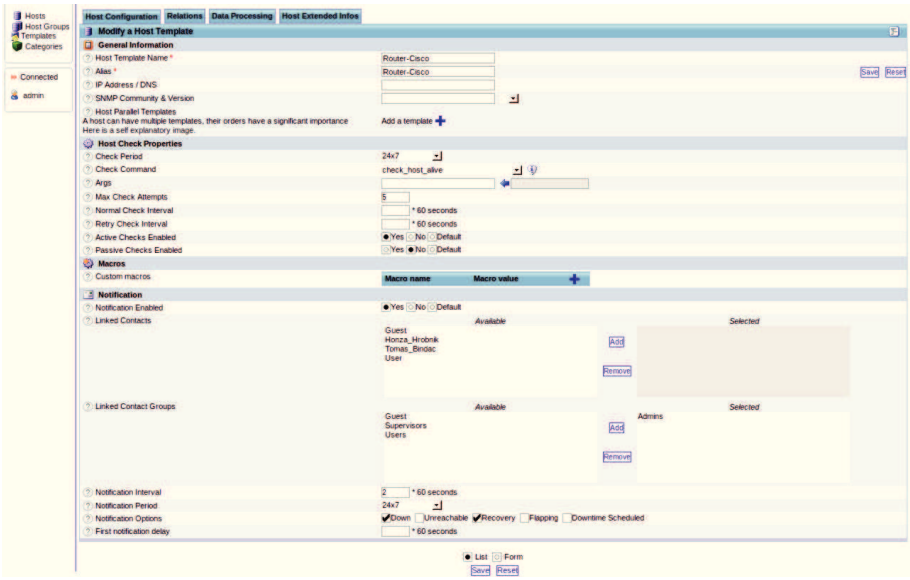
Můžeme si vytvořit skupinu kontaktů, kde přidáme více uživatelů, kterým má být zaslána informace, abychom je nemuseli vypisovat pokaždé zvlášť. Na obrázku 12 je zobrazeno nastavení pro skupinu uživatelů *Users*. Zde je nastaveno pouze jméno skupiny, *Alias* a samotní uživatelé, patřící do této skupiny.



Obrázek 12: Nastavení skupiny uživatelů

3.3.3 Templates

Dále si definujeme šablonu (*Templates*) kde si nastavíme jak často bude daný host dotazován, jakým příkazem to bude prováděno, kolikrát bude dotazován v případě výpadku, po jaké době se odešle chybová zpráva definovanému uživateli apod. Na obrázku 13 vidíme konfiguraci šablony pro routery Cisco. Název i *Alias* (zobrazeno v Nagiosu) *Router-Cisco*, dotazování hostů je nastaveno na 24 hodin 7 dní v týdnu, příkazem *check_host_alive* s maximálním počtem pěti dotazů a spuštěnou aktivní kontrolou hostů. To znamená, že pokud Nagios musí zkontrolovat stav nějakého hosta nebo služby bude k tomuto účelu vybrán plugin, díky kterému budou předány požadované informace. Jako poslední je nastavení chybových zpráv, které jsou samozřejmě zapnuty. Tyto zprávy jsou zasílány uživatelům ve skupině *Admins*. *Notification interval* je nastaven na hodnotu 2, to udává kolik časových jednotek (v našem případě dvě minuty) bude vyčkáváno před znovu odesláním zprávy danému kontaktu, že zařízení je stále *down* nebo *unreachable*.



Obrázek 13: Nastavení Templates

3.3.4 Host

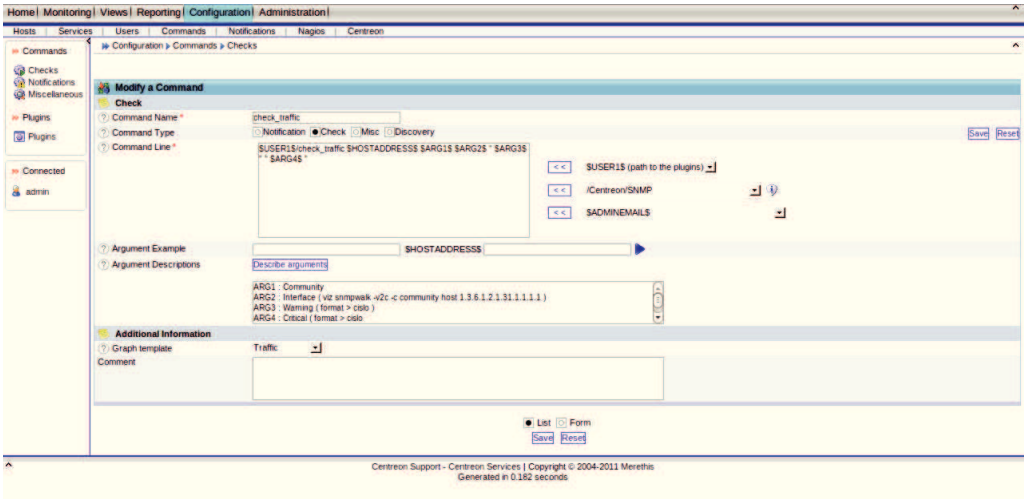
V záložce *Host* definujeme hosta, v tomto případě se jedná především o IP adresu daného zařízení. Jeho pojmenování, pod jakým bude uveden ve webovém rozhraní Nagiosu a určíme, kterou šablonu u hosta použijeme, at'nemusíme u každého hosta znovu vypisovat časové údaje o monitorování. Nastavení pro router B: jméno *RB*, *Alias* Cisco Router 2800 series, dále *IP adresa 192.168.0.2* a jako poslední je definice šablony, zde *Router-Cisco* viz výše.

3.3.5 Hostgroup

Opět můžeme také vytvořit skupinu hostů, například skupinu switchů, což nám usnadní další definování jejich služeb. Definice skupiny routerů má název *Routers* a identifikaci *Alias – All routers*. V záložce *Relations* vybereme hosty, které chceme k této skupině přidat, zde *RA-s0/1/0, RA-s0/1/1, RB a RC*.

3.3.6 Command

Dále si nadefinujeme příkaz (*Command*), jedná se o určení pluginů a jeho doplňujících argumentů, jako v případě SNMP komunity (*public, private* – záleží samozřejmě, jak jsme si ji sami nadefinovali), IP adresa monitorovaného zařízení a další argumenty, které jsou závislé přímo na použitém pluginu. Na obrázku 14 jsem vybral definici příkazu *check_traffic*. Typ příkazu je nastaven na *check*. Samotný příkaz obsahuje nejprve definici uživatele (jeho přístup k adresáři pluginů), zde $USER1$ za ním následuje samotný plugin *check_traffic* s jeho dodatečnými argumenty jako $HOSTADDRESS$ (IP adresa) a

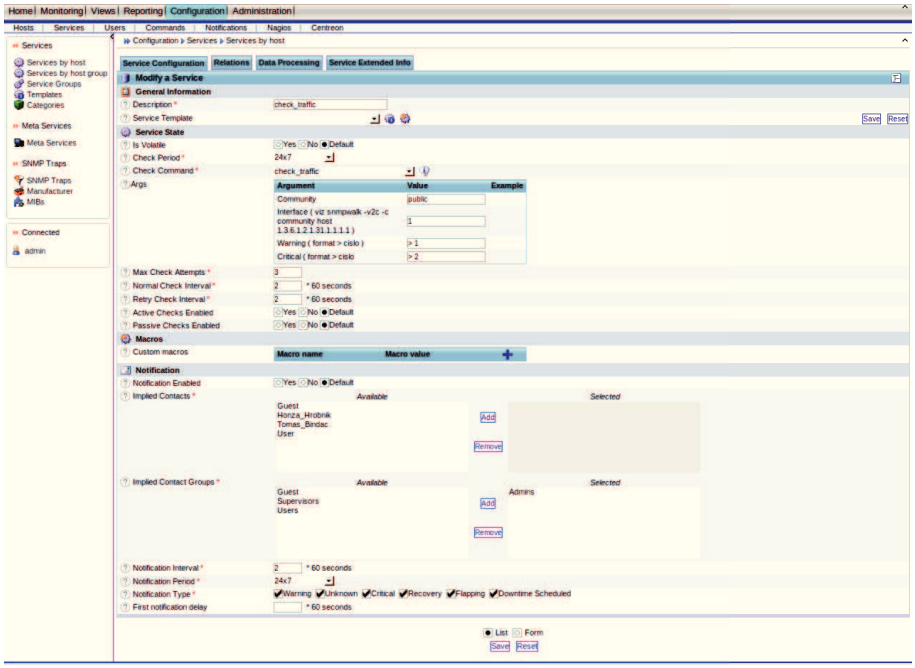


Obrázek 14: Nastavení Command

čtyři další argumenty $ARG1$ až $ARG4$ definované tak, že klikneme na pole *Describe arguments*, tím dojde k načtení těchto argumentů a můžeme k nim doplnit popis, který nám pomůže v pozdější definici služby. Jako poslední, avšak doplňkovou informací, je definice šablony pro grafy, zde *Traffic*.

3.3.7 Service

Nakonec vytvoříme samotnou službu (*Service*), kde zadáme náš vytvořený příkaz k monitorovaní viz výše, přesně zde definujeme jeho argumenty, nastavíme časové hodnoty pro monitorování a určíme, které hosty budeme danou službou monitorovat. Na obrázku 15 jsem vybral definici služby *check_traffic*, kterou bude host dotazován non-stop příkazem *check_traffic* a argumenty komunita *public*, rozhraní *1*, varovnou hodnotu > 1 (nad 1Mbit/s) a kritickou hodnotu > 2 (nad 2Mbit/s). Maximální počet dotazů, v případě, že je zařízení ve stavu *OK* je *3*. Časové rozmezí pravidelných kontrol služby, pokud je služba OK, nastavíme na dvě minuty. Jestliže není stav služby *OK*, je časové rozmezí pravidelných kontrol nastaveno také na dvě minuty. Předposledním parametrem je zasílání chybových zpráv *Notification*. Pro ilustraci jsme záměrně nevytvořili šablonu pro služby, abychom si ukázali její ruční nastavení. Nejprve samotné zasílání povolíme, následně vybereme uživatele, kterým budou oznámení zasílána, zde skupina uživatelů *Admins*. Interval oznámení je nastaven na dvě minuty a 24 hodin denně 7 dní v týdnu a odesíláním všech typů chybových zpráv viz výše. Jako poslední je nutné překliknout se na panel *Relations*, kde si vybereme k jakým hostům bude daná služba přidružena, zde *RA-s0/1/0, RA-s0/1/1, RB a RC*.



Obrázek 15: Nastavení Service

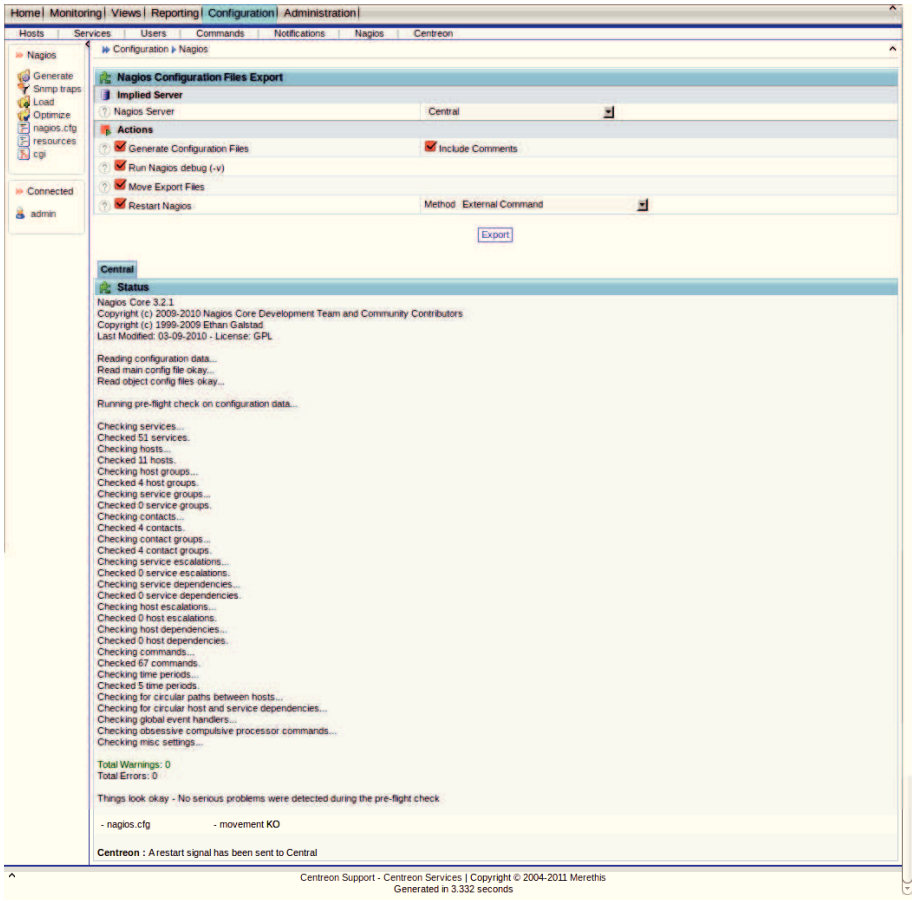
3.3.8 Export nastavení

Po dokončení vygenerujeme konfigurační soubory kliknutím na panel Nagios viz obrázek 16 a nastavením, jak již bylo zmíněno výše. Zaškrtneme veškeré položky v okně a v roletce *Method* zvolíme *External Command*. Pak už jen klikneme na *Export* a vše se nakopíruje do složky */etc/nagios*.

3.4 Monitoring (Odchytávání informací)

V této fázi máme systém nainstalován a nakonfigurován. Můžeme tedy přejít k monitoringu vlastní sítě. K tomu slouží webové rozhraní Nagios, přístupné z webového prohlížeče na adrese *http://localhost/nagios3* . V levém sloupci můžeme vidět samotné menu, skládající se z:

- *Tactical overview* – Sleduje monitorovací aktivitu z "ptačí perspektivy". To umožňuje rychle zobrazit výpadky sítě, statusy zařízení a služeb. Rozlišuje mezi problémy, které byly vyřešeny a problémy, které prozatím nebyly vyřešeny a potřebují pozornost. Tato položka je velmi užitečná v případě, že máme velké množství hostů/služeb a potřebujeme najednou zobrazit stav sítě.
- *Map* – Zobrazuje mapu sítě všech zařízení, které jsme na síti definovali. Tato mapa je ve formátu PNG. V horní liště je možné měnit zobrazení mapy např. *Collapsed tree* (zobrazuje zařízení na jednom řádku), *Circular* (zobrazuje zařízení okolo serveru) apod.



Obrázek 16: Export nastavení

- *Hosts* - Zobrazuje detailní stav (UP/DOWN) všech definovaných hostů, čas, kdy byla naposledy provedena kontrola zařízení, dobu, po kterou je zařízení spuštěno a podrobnější informace o jejich stavu.
- *Services* – Tato položka je téměř stejná jako předchozí *Hosts*, navíc zobrazuje ke každému hostu služby, které jsme definovali a v jakém stavu se nachází (OK, WARNING, CRITICAL, UNKNOWN, PENDING).
- *Host Groups* – Pokud jsme definovali skupiny hostů (není to nutnost), zobrazuje stav hostů v daných skupinách.
- *Service Groups* – Stejné jako v případě *Host Groups*.
- *Problems* – Tato položka zobrazuje výpis všech problémů pro jednotlivé hosty/služby, které na síti vznikly.
- *Availibiliy* – Slouží k zobrazení uživatelsky definovaných zpráv v určeném čase, o dostupnosti hosta, služby, skupiny hostů, nebo skupiny služeb.

- *Trends* – Umožňuje vytvářet grafy zařízení a služeb v libovolném čase. Tyto grafy zobrazují v procentech jak často byla daná zařízení či služby ve stavu OK, WARNING, UNKNOWN a CRITICAL.
- *Alerts history* – Tato položka slouží k zobrazení historie problémů všech hostů a služeb za určité období. Výstup je v podstatě podmnožinou informací, které jsou zobrazeny v log souboru.
- *Alerts summary* – Zobrazuje uživatelsky definovaný počet nejčastějších chyb vzniklých na síti.
- *Alerts histogram* – Zobrazuje histogram uživatelsky definovaných hostů/služeb v definovaném čase.
- *Notification* – Zobrazuje oznámení (chybovou zprávu) o hostech/službách, která byla zaslána daným uživatelům. Máme možnost filtrovat tento výpis na pouze určité typy oznámení, která chceme vidět (např. vypsat všechna oznámení služeb ve stavu CRITICAL).
- *Event log* – Zobrazuje jaké informace jsou zapsány do logovacího souboru, který je umístěn ve */var/log/nagios3/nagios.log*.
- *Comments* – Zde můžeme přidávat komentáře jednotlivým hostům/službám.
- *Downtime* – Umožňuje naplánovat kdy a na jak dlouho se má host/služba vypnout.
- *Process Info* – Zobrazuje povolené, či zakázané procesy na serveru. Umožňuje také spustit nebo zakázat různé příkazy (např. restartovat Nagios, vypnout oznámení, zastavit přijímání aktivních/pasivních kontrol, atd.)
- *Performace Info* – Zobrazuje v procentech kolikrát byli hosti/služby aktivně nebo pasivně sledováni.
- *Scheduling Queue* – Umožňuje zapnout/vypnout aktivní sledování jednotlivých služeb na zařízeních.
- *Configuration* – Vypíše uživatelem definovanou celkovou konfiguraci hosta/služby.

3.5 Skripty, pluginy a moduly

Pomocí skriptů můžeme tvořit například své vlastní šablony, či monitorovat hodnoty, které sami potřebujeme. Skripty mohou být napsány v jazyce Perl, Pythonu, PHP nebo bashi. Pokud známe čísla objektů OID, můžeme vytvořit své vlastní datové šablony k jiným zařízením. Každou šablonu tvoří dvojice programového kódu v Perlu, Pythonu, bashi nebo PHP s příponou .pl, .py, .sh nebo .php a skript

v jazyku XML. Pro lepší funkci monitorovacího systému můžeme doinstalovat přídavné moduly jako UCD-SNMP z balíku NET-SNMP[2], díky kterým můžeme získávat detailnější informace o procesoru, konkrétních procesech, paměti, disku, zátěži a souborech (včetně možnosti sledování logů pomocí regulárních výrazů). Zjednodušeně řečeno, abychom mohli pomocí MIB přeložit všechny potřebné hodnoty, které nám aktivní zařízení pomocí svého agenta odesílají.

Rozpis používaných pluginů či skriptů:

- *check_host_alive* – kontroluje, zda-li je zařízení aktivní (ve stavu UP)
- *check_uptime* – zjišťuje, jak dlouho je zařízení spuštěno
- *check_iterface* – kontroluje stav vybraných rozhraní, nejprve si spustíme příkaz *snmpwalk -v2c -c komunita host 1.3.6.1.2.1.31.1.1.1.1*, kde zjistíme jaké adresu jednotlivých rozhraní, kterou následně zapíšeme do definice příkazu viz kapitola 3.3.6
- *check_traffic* – kontroluje provoz sítě na zařízení (ukázka v příloze B)

Pluginy byly staženy ze stránky *exchange.nagios.org/directory/Plugins*, každý byl lehce poupraven cca. pěti řádky kódu.

Vlastní plugin zbývající volné místo na disku: Tento skript kontorluje volné místo na disku serveru (v mém případě notebooku) tím, že parsuje výstup z 'df /' a vrací Nagios výsledky jako stav OK, WARNING, CRITITCAL, nebo UNKNOWN.

```
#!/usr/bin/python
import re,sys,commands

#################
#Nastaveni promennych
command = "df /"
critical = 95.0
warning = 75.0
#################

#Porovnani regex
dfPattern = re.compile('[0-9]+')

#Vrat vyuzizi disku
diskUtil = commands.getstatusoutput(command)

#Rozdeli vracenou hodnotu v %
diskUtil = diskUtil[1].split()[11]

#Zkontoroluj shodu, pokud neni vypis v Nagiosu
#stav UNKNOWN (3)
```

[2]Stáhnout jej můžeme na stránce: *sourceforge.net/projects/net-snmp/files/net-snmp/5.7.1/net-snmp-5.7.1.tar.gz/download*, balík následně rozbalíme a pokračujeme v instalaci dle souboru README

```
matchobj = dfPattern.match(diskUtil)
if (matchobj):
   diskUtil = eval(matchobj.group(0))
else:
   print "stav UNKNOWN"
   sys.exit(3)

################################
#Slouzi k testovani diskUtil
#Odkomentuj nasledujici prikaz
#diskUtil = 98.0
################################

#Urcuje stav pro predani Nagiosu (cislo zaokrouhleno na dve
    desetinna mista)
#CRITICAL = 2
#WARNING = 1
#OK = 0
if diskUtil >= critical:
   print "Volne misto je ve stavu CRITICAL: '/' a je zaplneno
        z %.2f%%" % (float(diskUtil))
   sys.exit(2)
elif diskUtil >= warning:
   print " Volne misto je ve stavu WARNING: '/' a je zaplneno
        z %.2f%%" % (float(diskUtil))
   sys.exit(1)
else:
   print "Volne misto je ve stavu OK: '/' a je zaplneno z %.2
       f%%" % (float(diskUtil))
   sys.exit(0)
```

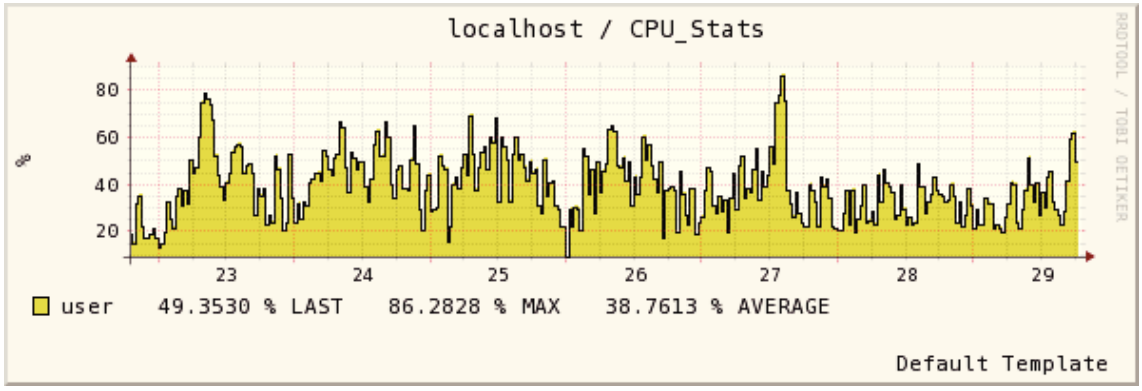
Nahoře definujeme příkaz *string* volaný v Pythonu *commands.getstatusoutput()* a nastavujeme kritické a varovné hodnoty. Tyto hodnoty v procentech můžeme libovolně měnit.

Další část kódu provedeme ve *stringu* rozdělení vrácené hodnoty v procentech. Pokud nedostaneme platnou hodnotu z *regex*, bude vrácena hodnota 3, což v Nagiosu znamená stav UNKNOWN.

Závěrečná část je rutinní porovnání vrácených hodnot s předdefinovanými pro CRITICAL a WARNING. Po porovnání vrátí kód správnou hodnotu Nagiosu.

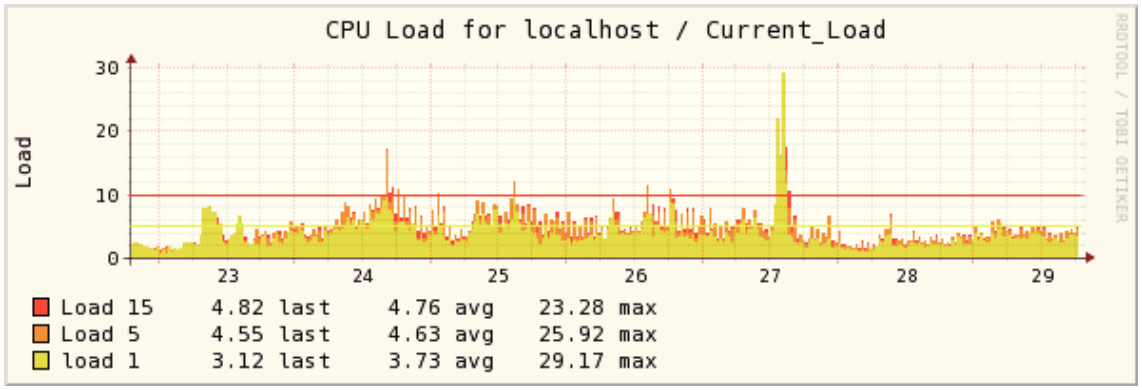
3.6 Grafy

K tomu, aby se nám vykreslovaly grafy můžeme použít několik způsobů. Jako první bych zmínil samotné rozhraní Centreon, kde při kliknutí na záložku *Views* a následně *Graph*, můžeme vidět grafy ke všem službám, které jsme v systému nakonfigurovali. V případě, že jsme si vytvořili skupiny hostů, budou stejně sloučeny i grafy a postupným proklikáním přes jednotlivá zařízení, rozhraní a samotnou službu, můžeme tyto grafy snadno zobrazit.



Obrázek 17: Graf statistiky CPU

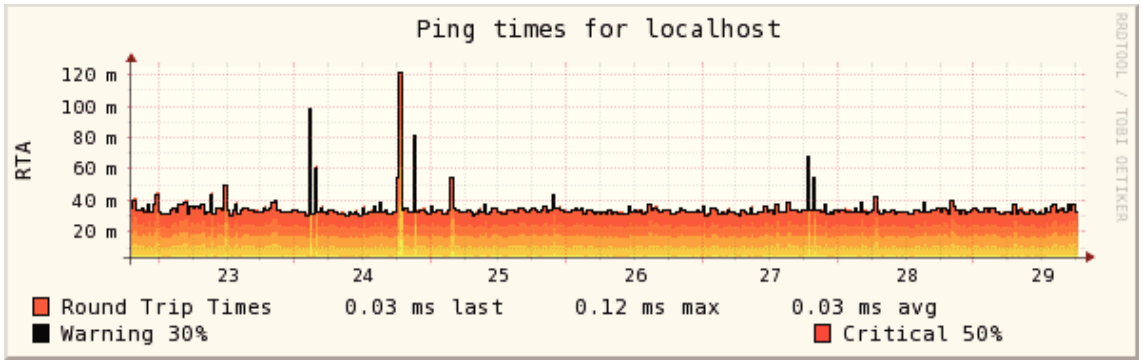
Na obrázku 17 vidímě graf pro CPU na Nagios serveru. Jedná se o jeho vytížení v procentech.



Obrázek 18: Graf statistiky CPU load

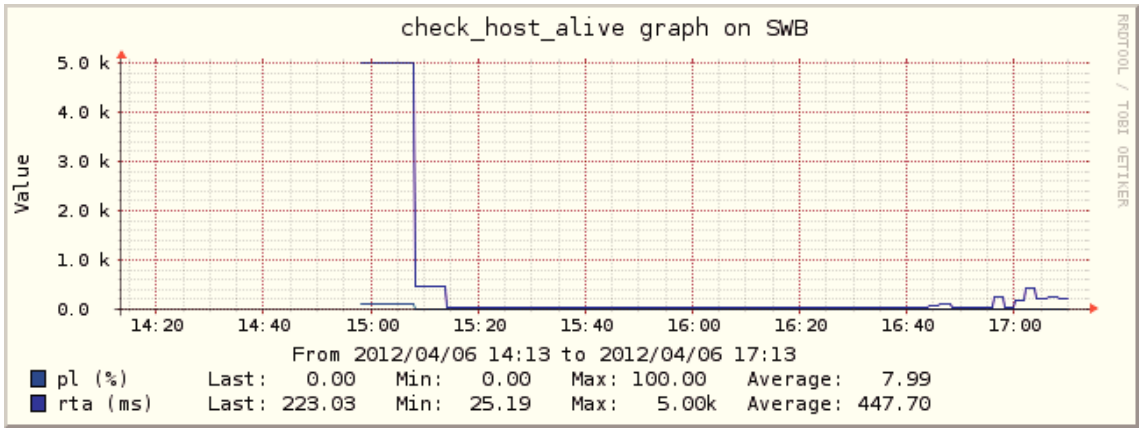
Na obrázku 18 vidíme zátěž procesoru na serveru Nagios. Jsou zde měřeny tři hodnoty pro průměrnou zátěž CPU v rozmezí jedné minuty, pěti minut a patnácti minut.

Na obrázku 19 vidíme statistiku Nagios v závislosti na příkaz *ping*. Jsou zde nastaveny prahové hodnoty pro *warning* na *30%* a *critical* na *50%*.



Obrázek 19: Graf statistiky ping

Graf služby check_host_alive pro zařízení SWBfig:CheckHost Graf na obrázku 20 udává odezvu na zařízení *SWB* pomocí příkazu *check_host_alive*.

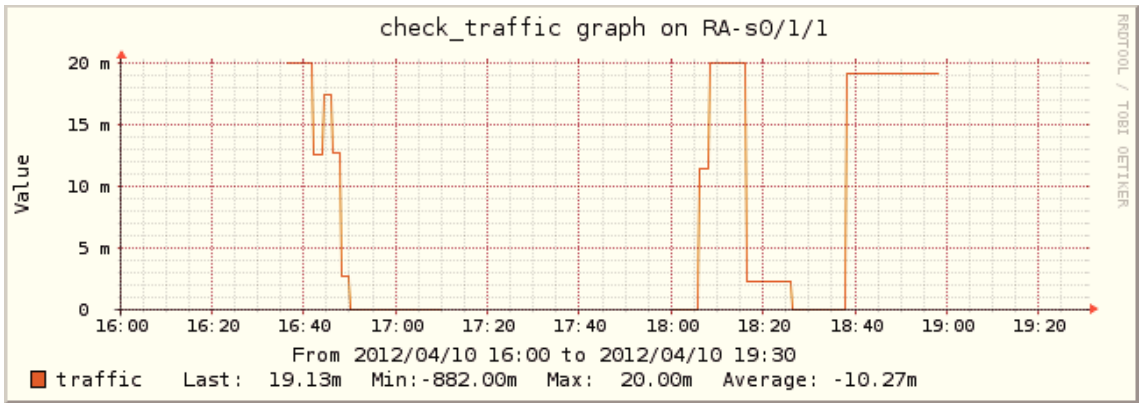


Obrázek 20: Graf služby check_host_alive pro zařízení SWB

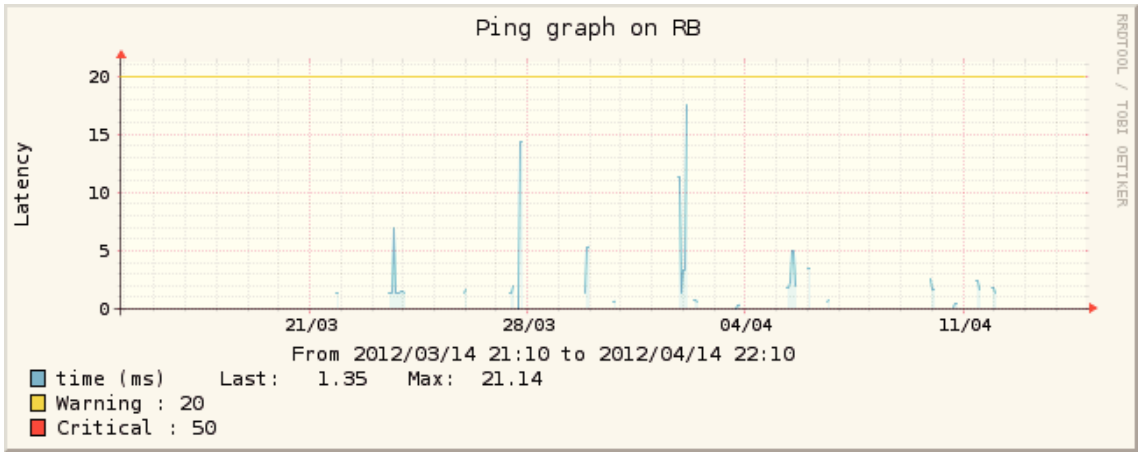
Obrázek 21 ukazuje provoz na zařízení *RA*, jeho rozhraní *s0/1/1* a příkazu *check_traffic*.

Graf na obrázku 22 zobrazuje měsíční graf příkazu *ping* na rozhraní *RB* v milisekundách. Prahové hodnoty jsou nastaveny, v případě *warning* na *20ms* a v případě *critical* na *50ms*.

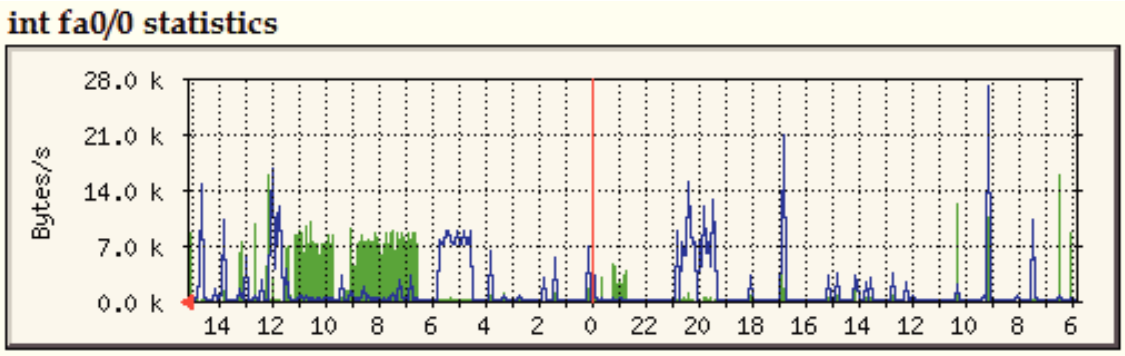
Graf na obrázku 23 zobrazuje provoz na zařízení SWA, tentokrát pomocí programu MRTG.



Obrázek 21: Graf služby check_traffic pro zařízení RA na rozhraní s0/1/1



Obrázek 22: Graf služby ping pro zařízení RB



Obrázek 23: MRTG graf provozu na zařízení SWA

4 Závěr

Cílem diplomové práce bylo navrhnout monitorovací systém pro středně velkou organizaci. Tuto práci jsem si vybral jako základ pro budoucí zaměstnání ve firmě s podobnou problematikou. Pro tento účel jsem, po analýze stavu existujících open-source systémů, vybral monitorovací systém Nagios, který vyniká svou širokou možností konfigurace a podává správci sítě hlavní informace o provozu v podobě statistik a grafů. Pomocí těchto síťových charakteristik je možné kontrolovat provoz a vykreslovat grafy vytíženosti sítě. Hlavním důvodem monitorování je však co nejrychlejší oprava problému v případě vzniklé chyby tak, aby byl opět zajištěn bezporuchový a bezpečný provoz sítě.

Nagios je velmi oblíbeným nástrojem, kolem kterého existuje velká komunita uživatelů, proto také vzniklo nepřeberné množství návodů na instalaci a konfiguraci, bohužel převážně ne dostatečně kvalitní nebo srozumitelné. Proto jsem se v této práci snažil instalaci a konfiguraci sepsat tak, aby byla použitelná například pro začínající správce sítě. Díky tomu si mohou rozšířit své znalosti v teoretické části a díky praktické části budou schopni systém nejen bezproblémově nainstalovat a nakonfigurovat, ale také ihned spustit na jejich síti a naplno tak využít funkce tohoto systému.

Pro ukázku funkčnosti systému, jsem vytvořil síť v laboratorních podmínkách, kde jsem se snažil použitými aktivními prvky simulovat reálnou podnikovou síť. V samotné konfiguraci sítě jsem vytvořil skript pro zobrazení stavu pevného disku na serveru viz B a také upravil několik dalších volně šiřitelných skriptů, pro nasazení v simulované síti. Z vygenerovaných grafů událostí na síti je zřejmé, že monitoring sítě byl úspěšně proveden. Takto vytvořený systém je možné použít v menších až středně velkých počítačových sítích.

Po vlastních zkušenostech bych zde uvedl ještě výhody a nevýhody spojené s tímto systémem. Mezi výhody řadím širokou oblast jeho použitelnosti, díky již mnohokrát zmiňovaným pluginům a skriptům. S jejich pomocí můžeme monitorovat prakticky veškeré potřebné hodnoty na síti. Také se jedná o kvalitní systém použitelný pro monitorování rozsáhlých sítí, čítající tisíce zařízení.

Mezi nevýhody bych zařadil fakt, že oproti jiným komerčním či nekomerčním systémům v prvotní instalaci neobsahuje žádné monitorovací možnosti, kromě pingu, takže veškeré hodnoty, které chceme monitorovat, si musíme nakonfigurovat sami. Dodatečný software Centreon nám tuto práci pouze zlehčuje v tom, že nemusíme hodnoty konfigurovat v textovém souboru a vkládat je do konfiguračního adresáře Nagiosu ručně. Jako další nevýhodu bych uvedl zdlouhavou a poměrně náročnou instalaci a konfiguraci celého systému. Ve webovém rozhraní systému navíc nelze

měnit nastavení sítě, pouze její funkčnost pasivně sledovat. K tomu nám ale může posloužit doplňkový nástroj Centreon.

5 Reference

[1] BIGELOW, Stephen J. *Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů.* Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. From technologies to solutions. ISBN 80-251-0178-9.

[2] BRUEY, Douglas. *SNMP: Simple? Network Management Protocol*. In: Rane Corporation [online]. 2005 [cit. 2011-12-01]. Dostupné z: www.rane.com/note161.html

[3] Cacti. [online]. 2004 [cit. 2012-04-24]. Dostupné z: www.cacti.net

[4] *ICTspecialista.cz: zápisky instalací pro správce i uživatele* [online]. 2012 [cit. 2012-03-02]. Dostupné z: www.ictspecialista.cz

[5] KLAŠKA, Luboš. *Další vývoj protokolu SNMP*. In: Svět sítí: Informace ze světa počítačových sítí [online]. 2000 [cit. 2012-04-01]. Dostupné z: www.svetsiti.cz/clanek.asp?cid=Dalsi-vyvoj-protokolu-SNMP-1562000

[6] KOCJAN, Wojciech. *Learning Nagios 3.0: a detailed tutorial to setting up, configuring, and managing this easy and effective system monitoring software.* Vyd. 1. Birmingham, U.K.: Packt Pub., c2008, 301 s. From technologies to solutions. ISBN 978-1-84719-518-0.

[7] KOCMAN, Jiří. *Jak na démona Cron*. In: Interval.cz [online]. 2002 [cit. 2012-02-03]. Dostupné z: interval.cz/clanky/jak-na-demona-cron/

[8] KOZÁK, Milan. *Domácí počítačová síť - 9.* In: Linuxexpres [online]. 2008 [cit. 2012-03-07]. Dostupné z: www.linuxexpres.cz/praxe/domaci-pocitacova-sit-9

[9] MATÚŠŮ, Jindřich. *Monitorování stavu rozsáhlých sítí*. Zlín, 2008. Diplomová. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Ing. Tomáš Dulík.

[10] OpenNMS. [online]. 2002 [cit. 2012-04-27]. Dostupné z: www.opennms.org

[11] PUŽMANOVÁ, Rita. *Řízení komunikačních sítí na bázi protokolů CMIP (OSI) a SNMP (TCP/IP)*. Praha, 1992. Kandidátská disertační práce. ČVUT - Katedra inženýrské informatiky. Vedoucí práce doc. Ing. Petr Moos, CSc.

[12] ROSE, Marshall T a Keith MCCLOGHRIE. *How to manage your network using SNMP: the networking management practicum*. Englewood Cliffs, N.J.: PTR Prentice Hall, c1995, 549 s. ISBN 01-314-1517-4.

[13] RRDTool. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2011-11-02]. Dostupné z: cs.wikipedia.org/wiki/RRDTool

[14] SCHRODER, Carla. *Linux: kuchařka administrátora sítě.* Vyd. 1. Brno: Computer Press, 2009, 596 s. ISBN 978-80-251-2407-9.

[15] SCHUBERT, Max. *Nagios 3 enterprise network monitoring: including plug-ins and hardware devices.* Vyd. 1. Překlad Petr Matějů. Burlington, MA: Syngress Pub., c2008, 348 s. From technologies to solutions. ISBN 15-974-9267-1.

[16] Technet Microsoft. How SNMP Works [online]. 2003 [cit. 2011-10-20]. Dostupné z: technet.microsoft.com/en-us/library/cc783142(v=ws.10).aspx

[17] Zabbix. [online]. 2001 [cit. 2012-04-27]. Dostupné z: www.zabbix.com

[18] *Sdělovací technika: telekomunikace - elektronika - multimédia.* Praha: Petr Beneš v nakladatelství Sdělovací technika s. r. o. ISSN 0036-9942. Dostupné z: www.ist-lobster.org/publications/articles/sdel_tech.pdf.

A Konfigurační soubory

Statické směrování: Router A

```
minicom -s
  -> nastavit port na ttyS0 a rychlost na 9600kb
  -> enable
  -> conf t
  -> hostname RA
  -> int s0/1/0
    -> ip address 192.168.0.1 255.255.255.0
    -> no sh
    -> clock rate 64000
    -> ip route 172.16.1.0 255.255.255.0 192.168.0.2
    -> exit
  -> int s0/1/1
    -> ip address 192.168.1.1 255.255.255.0
    -> no sh
    -> clock rate 64000
    -> ip route 172.16.2.0 255.255.255.0 192.168.1.2
    -> exit
-> int fa0/0
    -> ip address 172.16.0.1 255.255.255.0
    -> no sh
    -> exit

SNMP:
  -> conf t
    -> snmp-server community public RO
    -> snmp-server community private RW
    -> exit
  -> snmp-server host 172.16.0.4 version 1 private
  -> snmp-server enable traps snmp linkdown linkup coldstart
     warmstart
-> write memory
```

Router B

```
minicom -s
  -> nastavit port na ttyS0 a rychlost na 9600kb
-> enable
  -> conf t
  -> hostname RB
  -> int s0/1/0
    -> ip address 192.168.0.2 255.255.255.0
    -> no sh
    -> clock rate 64000
    -> ip route 172.16.0.0 255.255.255.0 192.168.0.1
    -> exit
```

```
-> int fa0/0
     -> ip address 172.16.1.1 255.255.255.0
     -> no sh
     -> exit

SNMP:
  -> conf t
     -> snmp-server community public RO
     -> snmp-server community private RW
     -> exit
  -> snmp-server host 172.16.0.4 version 1 private
  -> snmp-server enable traps snmp linkdown linkup coldstart
      warmstart
-> write memory
```

Router C

```
minicom -s
  -> nastavit port na ttyS0 a rychlost na 9600kb
  -> enable
-> conf t
  -> hostname RC
  -> int s0/1/0
     -> ip address 192.168.1.2 255.255.255.0
     -> no sh
     -> clock rate 64000
     -> ip route 172.16.0.0 255.255.255.0 192.168.1.1
     -> exit
-> int fa0/0
     -> ip address 172.16.2.1 255.255.255.0
     -> no sh
     -> exit

SNMP:
  -> conf t
     -> snmp-server community public RO
     -> snmp-server community private RW
     -> exit
  -> snmp-server host 172.16.0.4 version 1 private
  -> snmp-server enable traps snmp linkdown linkup coldstart
      warmstart
-> write memory
```

Switch A

```
minicom -s
-> enable
-> conf t
-> hostname SWA
automatické nastavení správného času (kvůli logům)
```

```
-> conf t
-> ntp server 10.0.0.50
správná časová zóna
-> conf t
-> clock timezone GTM 1
čas pro logy a debug
-> conf t
-> service timestamps debug datetime
  -> service timestamps log datetime
nastavení IP adresy
-> conf t
-> int vlan 1
  -> ip address 172.16.0.2 255.255.255.0
  -> no sh
SNMP:
-> conf t
     -> snmp-server community public RO
     -> snmp-server community private RW
     -> snmp-server host 172.16.0.4 version 1 private
     -> snmp-server enable traps snmp linkdown linkup
        coldstart warmstart
  -> write memory
```

Switch B

```
minicom -s
-> enable
-> conf t
-> hostname SWB
automatické nastavení správného času (kvůli logům)
-> conf t
-> ntp server 10.0.0.50
správná časová zóna
-> conf t
-> clock timezone GTM 1
čas pro logy a debug
-> conf t
-> service timestamps debug datetime
  -> service timestamps log datetime
nastavení IP adresy
-> conf t
-> int vlan 1
  -> ip address 172.16.1.2 255.255.255.0
  -> no sh

SNMP:
-> conf t
     -> snmp-server community public RO
     -> snmp-server community private RW
     -> snmp-server host 172.16.0.4 version 1 private
```

```
      -> snmp-server enable traps snmp linkdown linkup
         coldstart warmstart
    -> write memory
```

Switch C

```
minicom -s
-> enable
-> conf t
-> hostname SWC
automatické nastavení správného času (kvůli logům)
-> conf t
-> ntp server 10.0.0.50
správná časová zóna
-> conf t
-> clock timezone GTM 1
čas pro logy a debug
-> conf t
-> service timestamps debug datetime
  -> service timestamps log datetime
nastavení IP adresy
-> conf t
-> int vlan 1
  -> ip address 172.16.2.2 255.255.255.0
  -> no sh
SNMP:
-> conf t
     -> snmp-server community public RO
     -> snmp-server community private RW
     -> snmp-server host 172.16.0.4 version 1 private
     -> snmp-server enable traps snmp linkdown linkup
        coldstart warmstart
  -> write memory
```

OSPF - avšak není nutné

```
RA - ospf
router ospf 1
  -> network 192.168.0.0 0.0.0.255 area 0
  -> network 192.168.1.0 0.0.0.255 area 0
  -> exit

RB - ospf
router ospf 1
  -> network 192.168.0.0 0.0.0.255 area 0
  -> exit

RC - ospf
router ospf 1
```

```
-> network 192.168.1.0 0.0.0.255 area 0
-> exit
```

PC1

```
ifconfig eth0 172.16.0.3 netmask 255.255.255.0
route add default gw 172.16.0.1

SNMP:
apt-get install snmp snmpd
mc /etc/default/snmpd - smazat IP z SNMPDOPTS
service snmpd restart

kontrola funkčnosti:
snmpwalk -c public -v1 172.16.0.3
```

PC2

```
ifconfig eth0 172.16.1.3 netmask 255.255.255.0
route add default gw 172.16.1.1

SNMP:
apt-get install snmp snmpd
mc /etc/default/snmpd - smazat IP z SNMPDOPTS
service snmpd restart

kontrola funkčnosti:
snmpwalk -c public -v1 172.16.1.3
```

PC3

```
ifconfig eth0 172.16.2.3 netmask 255.255.255.0
route add default gw 172.16.2.1

SNMP:
apt-get install snmp snmpd
mc /etc/default/snmpd - smazat IP z SNMPDOPTS
service snmpd restart

kontrola funkčnosti:
snmpwalk -c public -v1 172.16.2.3
```

B Skript pro kontrolu provozu na síti – check_traffic

```
#!/bin/sh

#----Help------------------#

if [ "$1" = "help" -o ! "$#" -gt "4" ]; then
  echo -e "\nCheck traffic usage of an interface\n";
  echo -e "--------------------------------------------------
     ";
  echo -e "Usage: ./check_snmp_traffic <host> <snmp-community
     > <if-number> <warn> <crit>";
  echo -e "
     ------------------------------------------------\n";
  echo -e "<host>              Hostname or IP Address";
  echo -e "<snmp-community>    the snmp community string";
  echo -e "<if-number>         Interface number, use snmpwalk
     to find yours:";
  echo -e "";
  echo -e "       snmpwalk -v2c -c community host
     1.3.6.1.2.1.31.1.1.1.1";
  echo -e "       IF-MIB::ifName.1 = STRING: Fa0";
  echo -e "                This -^- is the interface-number";
  echo -e "";
  echo -e "<warn>              warning if bc expression return
      1";
  echo -e "<crit>              critical if bc expression
     return 1";
  echo -e ""
  echo -e "Examples: "
  echo -e "# ./check_snmp_traffic localhost private 1 \" > 1
     \" \" > 2 \"";
  echo -e "WARNING if iface # 1 trafic > 1 Mbit/s, CRITICAL
     if trafic > 2 Mbits/";
  echo -e ""
  echo -e "# ./check_snmp_traffic localhost private 10 \" <
     11 \" \" < 5 \"";
  echo -e "WARNING if iface # 10 has trafic < 11 Mbit/s,
     CRITICAL if trafic < 5 Mbits/";
  echo -e ""

  exit 0;

fi

#----variables-set-at-runtime-----------#

sleep=5
host=$1
```

```
snmpstring=$2
interface=$3
warn=$4
crit=$5

snmp_ifin="1.3.6.1.2.1.31.1.1.1.6"
snmp_ifout="1.3.6.1.2.1.31.1.1.1.10"

# last value
traflastIn='snmpget -v2c -c $snmpstring $host $snmp_ifin.
   $interface |awk {'print $4'}'
traflastOut='snmpget -v2c -c $snmpstring $host $snmp_ifout.
   $interface |awk {'print $4'}'

sleep $sleep

# current value
trafbyteIn='snmpget -v2c -c $snmpstring $host $snmp_ifin.
   $interface |awk {'print $4'}'
trafbyteOut='snmpget -v2c -c $snmpstring $host $snmp_ifout.
   $interface |awk {'print $4'}'

# echo "traflastIn: $traflastIn"
# echo "trafbyteIn: $trafbyteIn"
#
# echo "traflastOut: $traflastOut"
# echo "trafbyteOut: $trafbyteOut"

#----calculation-In----------------------#

if [ $trafbyteIn -gt $traflastIn ]; then
  trafdiffIn=$(echo " $trafbyteIn - $traflastIn " | bc)

elif [ $trafbyteIn -lt $traflastIn ]; then
  # this counter cannot be reset, unless the system was
     restarted
  # so no calculation here - we assume the current value IS
     the diff
  trafdiffIn=$trafbyteIn

fi

#----calculation-Out---------------------#

if [ $trafbyteOut -gt $traflastOut ]; then
  trafdiffOut=$(echo " $trafbyteOut - $traflastOut" | bc)

elif [ $trafbyteOut -lt $traflastOut ]; then
```

```
    # this counter cannot be reset, unless the system was
       restarted
    # so no calculation here - we assume the current value IS
       the diff
    trafdiffOut=$trafbyteOut

fi

#----human-readable-for-output-----------#

trafmbIn=`echo "scale=2; ( $trafbyteIn - $traflastIn ) * 8 /
   $sleep / 1024 / 1024 " | bc`
trafmbOut=`echo "scale=2; ( $trafbyteOut - $traflastOut ) * 8
    / $sleep / 1024 / 1024 " | bc`
trafmb=`echo "scale=2; $trafmbIn + $trafmbOut" | bc`
trafmb1=`echo "$trafmb / 1" | bc`


# echo "trafmbIn: $trafmbIn"
# echo "trafmbOut: $trafmbOut"
# echo "trafmb: $trafmb"
# echo "trafmb1: $trafmb1"
#

#----write-values-for-next-run-----------#

# if any variable is 0 or has no value at all better do
   nothing and quit unknown
for X in "$trafbyteIn" "$trafbyteOut"
do
  if [ -z $X ]; then
      echo "Traffic UNKNOWN - $trafmb Mb/s in Sum|traffic=
         $trafmb;$warn;$crit;0; In=$trafmbIn;;;0; Out=
         $trafmbOut;;;0;"
      exit 3
  fi
  if [ $X = 0 ]; then
      echo "Traffic UNKNOWN - $trafmb Mb/s in Sum|traffic=
         $trafmb;$warn;$crit;0; In=$trafmbIn;;;0; Out=
         $trafmbOut;;;0;"
      exit 3
  fi
done

# echo $trafbyteIn $trafsumIn $trafbyteOut $trafsumOut >
   $tmpfile

#----output-and-exit-code---------------#
```

```
exprwarn=` echo $trafmb1 $warn | bc `
exprcrit=` echo $trafmb1 $crit | bc `

if [ $exprcrit -eq 1 ]; then
  echo "Traffic CRITICAL - $trafmb Mb/s in Sum|traffic=
     $trafmb;$warn;$crit;0; In=$trafmbIn;;;0; Out=$trafmbOut
     ;;;0;"
  EXIT=2
elif [ $exprwarn -eq 1 ]; then
  echo "Traffic WARNING - $trafmb Mb/s in Sum|traffic=$trafmb
     ;$warn;$crit;0; In=$trafmbIn;;;0; Out=$trafmbOut;;;0;"
  EXIT=1
elif [ $exprwarn -eq 0 -a $exprcrit -eq 0 ]; then
  echo "Traffic OK - $trafmb Mb/s in Sum|traffic=$trafmb;
     $warn;$crit;0; In=$trafmbIn;;;0; Out=$trafmbOut;;;0;"
  EXIT=0
else
  echo "Traffic UNKNOWN - $trafmb Mb/s in Sum|traffic=$trafmb
     ;$warn;$crit;0; In=$trafmbIn;;;0; Out=$trafmbOut;;;0;"
  EXIT=3
fi

exit $EXIT
```